NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEGRATION OF THE NAVY TACTICAL ENVIRONMENTAL DATABASE SERVICES WITH THE JOINT EFFECTS MODEL

by

Victor B. Ross III

December 2003

Thesis Advisor:	Neil Rowe
Second Reader:	Carlyle Wash

Approved for public release, distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** December 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Integration of the Navy Tactical Environmental Database Service with the Joint Effects Model

5. FUNDING NUMBERS

6. AUTHOR (S) Victor B. Ross III, LCDR, USN

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release, distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(maximum 200 words)***

The Oceanographer of the Navy is responsible for the maintenance and distribution of the "4-D cube" of environmental data, the Virtual Natural Environment, using an object oriented database and distribution system, Tactical Environmental Database Services (TEDServices). The new military dispersion modeling capability within the military is called the Joint Effects Model (JEM), and has to have an interface created to allow inclusion of weather data in JEM. This thesis utilizes TEDServices using web protocols to query for available data, and then retrieves the required meteorology data. The software creates a specifically formatted file to be used in JEM. It is now fully functional and submitted to Space and Warfare Command for inclusion in JEM. Much of the testing was to ensure that the data are available and within the reasonable meteorological standards. The thesis also suggests additional changes that should be made to TEDServices to make it more capable of storing and serving environmental data.

14. SUBJECT TERMS REA, JEM, Dispersion Model, TEDServices, HPAC, Mesoscale Model			**15. NUMBER OF PAGES** 112
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 **Standard Form 298 (Rev. 2-89)**
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release, distribution is unlimited

INTEGRATION OF THE NAVY TACTICAL ENVIRONMENTAL DATABASE SERVICE WITH THE JOINT EFFECTS MODEL

Victor B. Ross III
Lieutenant Commander, United States Navy
B.S., Florida Institute of Technology, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
December 2003

Author: Victor B. Ross III

Approved by: Dr. Neil Rowe
Thesis Advisor

Dr. Carlyle H. Wash
Second Reader

Dr. Peter Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Oceanographer of the Navy is responsible for the maintenance and distribution of the “4-D cube” of environmental data, the Virtual Natural Environment, using an object oriented database and distribution system, Tactical Environmental Database Services (TEDServices). The new military dispersion modeling capability within the military is called the Joint Effects Model (JEM), and has to have an interface created to allow inclusion of weather data in JEM. This thesis utilizes TEDServices using web protocols to query for available data, and then retrieves the required meteorology data. The software creates a specifically formatted file to be used in JEM. It is now fully functional and submitted to Space and Warfare Command for inclusion in JEM. Much of the testing was to ensure that the data are available and within the reasonable meteorological standards. The thesis also suggests additional changes that should be made to TEDServices to make it more capable of storing and serving environmental data.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

There are a number of people who made this thesis possible, and while painful at times I hope that it improves operational procedures in the future. First, I must thank my section, Nick, Jeff, Claude, Juan, and honorary member Dave Brown, for all the help getting through two degree programs at the same time. I also want to thank my advisors Chuck Wash and Neil Rowe. Their taking the time to understand the other half of each thesis made the task possible. Of course, the support from SPAWAR was critical to get this project started. Thanks to John Shea for making it all possible. Along the way I was given technical and morale support in many areas. Thanks to the following for keeping me on track and moving forward: Mike Frost, John Cook, Larry Pheagley, Steve Hammett, Dan Geiszler, Roy Ladner, and Carl Baribault. Finally, there was another part of life that I can't dismiss. While not in class, I had to find diversions to make life worth living. My time in Monterey would not have been fulfilling without my time spent in the Trident Room for METOC Thursdays. A little closer to home, I have to thank the Bulldog Pub, Barbary Coast Theater, and Monterey Bay Aquarium for giving me something to do with my free time while here in Monterey.

I. INTRODUCTION

A. BACKGROUND

The new Operational Concept published by the Oceanographer of the Navy (N096) proposes a new way of collecting and disseminating meteorological and oceanographic data to military forces. Its goal is "Empower our operating forces to dominate the battle space through superior understanding and exploitation of the natural environment of the ocean and atmosphere" (Oceanographer of the Navy, 2002). To enable this concept to become a reality, the "4D Cube" is proposed. It is defined as "a virtual entity of geospatially referenced data, information and knowledge used to support interoperable nodes/systems" (Oceanographer of the Navy, 2002). The 4D cube is basically a 3 dimensional representation of the world with the added time dimension. The knowledge base that will house the 4D cube will be called the Virtual Natural Environment (VNE).

This knowledge base will be implemented using Tactical Environmental Database Services (TEDServices), a replicated database running at Meteorology and Oceanography (METOC) locations around the globe. This replicated system of data allows for continuous data retrieval at remote locations if there are breaks in communications which could preclude forecasters from completing their jobs. This includes the next generation of dispersion modeling software for the Department of Defense (DOD) called the Joint Effects Model (JEM) (Integrated Chemical and Biological Defense Research, Development and Acquisition Plan, 2003). The software needs

access to the most recent meteorological data sets stored in TEDServices to run simulations on the effects of Weapons of Mass Destruction (WMD) and Weapons of Mass Effect (WME). Dispersion models are used to simulate the movement and dispersion of the nuclear, biological, and chemical (NBC) agents based upon the model prediction of the atmospheric conditions (Johnson-Winegar, 2003). Biological and chemical attacks are always a concern for deployed military forces, but the Navy must also contend with a possible strike that could cause nuclear dispersion. Since any strike against troops using an NBC agent is potentially lethal, the interface back to the METOC data is a critical system path.

B. THESIS WORK

TEDServices and JEM require a new database storage capability and network access ability. JEM is a mathematical model consisting of both Java and Fortran code designed to calculate the dispersion of NBC agents. JEM is available both as a Web-enabled and stand-alone application. Each implementation needs direct access to the current meteorological data stored in TEDServices. Since most computer security restrictions only allow Web based queries, this thesis creates an access capability to be used by JEM to access TEDServices using the Web access protocol on ports 80 or 443. This access will normally be within a localized intranet if possible, but can also be across the larger network infrastructure if there is no local TEDServices. It also allows for placing data created by JEM back into a local or centralized TEDServices database. This allows for the data to be displayed as part

of the Common Relevant Operating Picture (CROP) to Joint Forces around the world. This should support the areas of the Operational Concept as highlighted in Figure 1.

C. BENEFIT GAINED

Current WMD/WME data is specifically run on dedicated platforms, and the output is only displayed as independent graphics for decision makers. Since all operational dispersion models must have meteorological boundary conditions to run (Defense Threat Reduction Agency, 2003), the proposed implementation of JEM and TEDServices will allow for increased availability of the dispersion modeling capability as well as the dissemination of the output from the dispersion model.

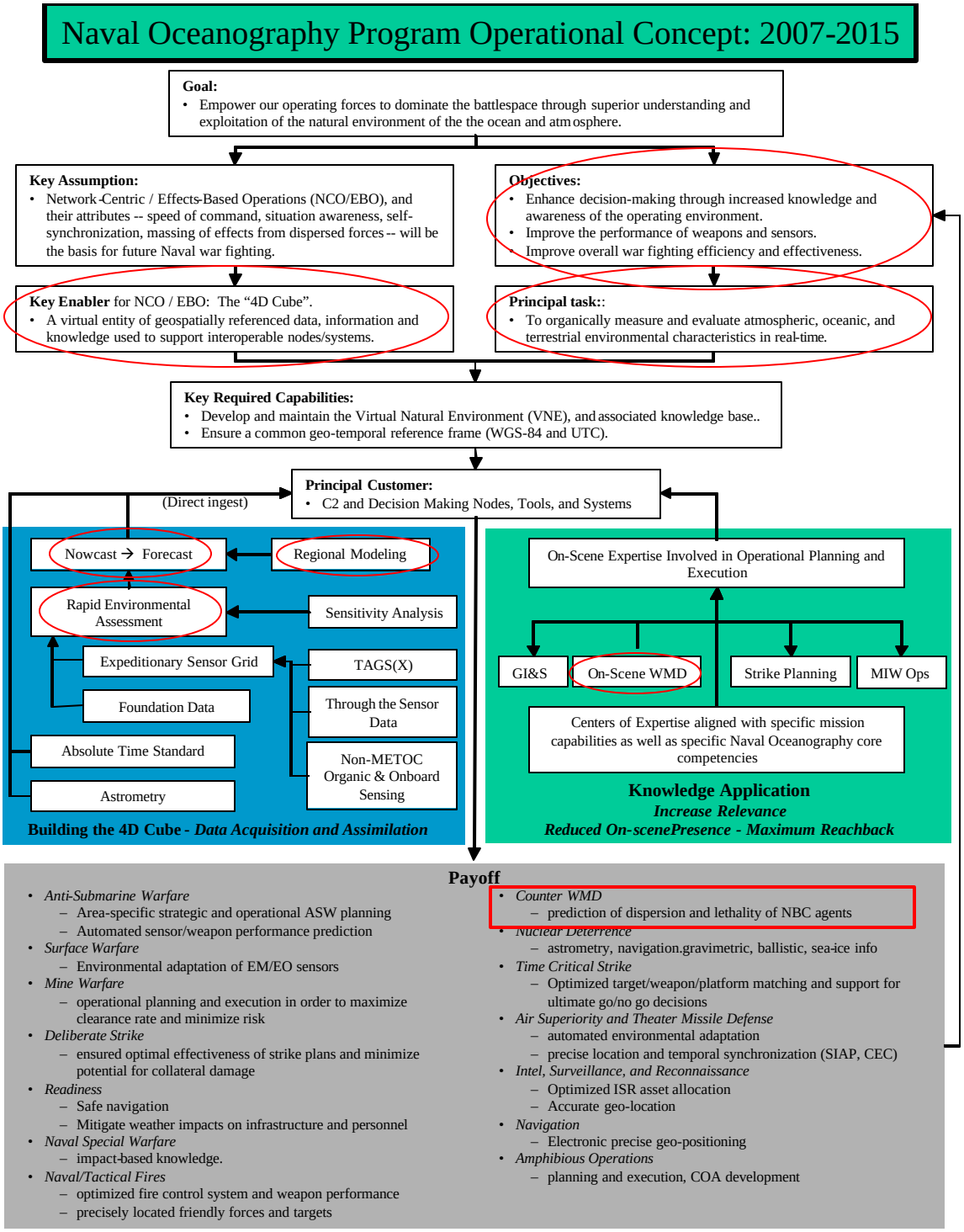

Figure 1. N096 Operational Concept (After Oceanographer of the Navy, 2002)

II. WEATHER DATABASES AND DATA TRANSMISSION

A. BACKGROUND

Many reasons for using a database to store weather information may not be obvious to the non-meteorologist. Many issues associated with weather data collection and dissemination can only be effectively solved using databases. The first is the actual collection locations which are scattered around the world and orbiting on satellites. This creates thousands of possible data points over the course of a day. Second is the timing of data collection and use. Weather data is highly perishable with valid use times ranging from seconds to hours and is frequently collected at irregular intervals. The other major issue can be associated with the physical size of the data files themselves. Each of the previously listed data sources give only the current data collected from the atmosphere, but the forecasting of atmospheric conditions can create even larger data sets. Binary forecast data files representing only a single parameter at a single atmospheric level at a single time are currently around 45Kb, and standard production models such as the Navy Operational Global Atmospheric Prediction System (NOGAPS) can have over 100 atmospheric parameters, 30 levels, and over 32 times. In comparison to that, a single observation can be less than 200 characters, but there are thousands issued every day. The actual forecast usually deals with a specific subset of the binary data and a superset of observations. Bandwidth limitations for deployed units make it impossible to access all the available data sets.

For a database to be used to store METOC data, it has to be able to ingest geographic coordinates, parameters, levels, and times. Specific subsets of requested data from all the data stored in the database are then sent to the user in a usable format as defined by both the military and the World Meteorological Organization (WMO). This must include the different in situ and remotely sensed observations as well as forecast information that fits within the request area and time. Since METOC needs vary drastically according to the work being proposed, a database must store and retrieve the data rapidly from the larger data store.

B. CURRENT PROCEDURES

The current system for disseminating weather data in the U.S. Navy uses a hybrid Informix database, Tactical Environmental Data Server (TEDS), and a transmission system on network port 80 or 443 called METCAST. The primary database is located at Fleet Numerical Meteorology and Oceanography Center (FNMOC) in Monterey, CA. Regional centers located around the world, as well as carriers and amphibious landing ships, have smaller versions installed. Each of the smaller versions can pull data from the other servers as well as adding their own localized data to the database.

While there are still numerous versions deployed around the world, over the past seven years the three different versions of TEDS have been consolidated into a single version. The different versions were developed from the same original framework, but the data stores were

modified to contain specific data structures to accommodate separate developmental programs. The first operational version of Single TEDS was sent to Hawaii for final testing in early 2003. The single version of TEDS has helped to reduce the overall expense of maintaining the systems, but since this system uses a proprietary database, the maintenance costs are still very high.

C. FUTURE WORK

In order to allow for more copies of a METOC database at all available locations, a new data dissemination system, TEDServices, is being created. This Java-based, object-oriented database, Ozone (Ozone Database Project), is open-source and can be replicated many times within Navy activities without charge. Much like METCAST, the data dissemination portion of the system uses an open-source Web server, Apache-Tomcat, to allow data pulls over accepted Navy ports such as 80 and 443. The open-source nature of the TEDServices database makes the total cost of development significantly lower, and there is no increased cost for deploying a larger network of databases around the Navy. This is particularly important since the databases will have to be deployed to enable Rapid Environmental Assessment (REA), and constant communications with deployed units is not guaranteed. The REA process is new to military applications, and will improve forecasting by looking only at short term forecasts utilizing the most current data sources. While most modeling is done on a set schedule, the REA process is designed to reassess the environment on a shorter and possibly irregular time scale. Having an on-scene METOC database allows the deployed units

to continue to fulfill obligated REA duties when communications are interrupted. Each of the METOC databases will need to ingest local information and disseminate that information to weapons and modeling systems within the deployed network.

D. THESIS WORK

This thesis enables sites to request weather data from TEDServices for dispersion modeling. This will allow JEM or the current dispersion model, Hazard Prediction and Assessment Capability (HPAC), to retrieve the required fields to model the dispersion from a WMD/WME. Since both JEM and TEDServices are in early development, there is no interface between the two applications. This thesis will build an interface as part of JEM. The interface is written in Java 1.4.1, and will extract data from the current beta version 2.2 of TEDServices (Naval Research Laboratory – Stennis Space Center, 2003).

Data extraction from the database requires “cutting” 3-dimensional subsections and reallocating them to follow the MEDOC grid specification required for JEM or HPAC. MEDOC is a specific meteorological data format created by the Defense Threat Reduction Agency (DTRA) for use in dispersion modeling. This requires examining map projection, grid density, grid boundaries, sigma levels, topography, and grid de-staggering to make the required extraction usable by JEM. It will also have to check the database for available models that cover the same geography and time constraints in the requested area, choosing among different models, geographic coverages, times, and data

resolutions. The next proposed plan is to create a secondary database containing only the VNE. This will only store a current time, best model and resolution for the entire globe. The decision as to how the models will be selected to create the VNE has not yet been made, so this thesis will use the Navy's Coupled Ocean-Atmosphere Mesoscale Prediction System – On Scene (COAMPS-OS) as a first attempt to represent the best atmospheric forecast used in the VNE.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SOFTWARE DESIGN AND OPERATING PROCEDURES

A. SOFTWARE DESIGN

The current methods for METOC data movement will require changes to be successful in the future. This thesis addresses a single problem within that movement pattern. With the current tools, data must be passed in and out of various databases. The overall data flow is shown in Figure 2, showing how TEDS and TEDServices must be queried to get the data in and out of the TEDServices database for use by JEM.

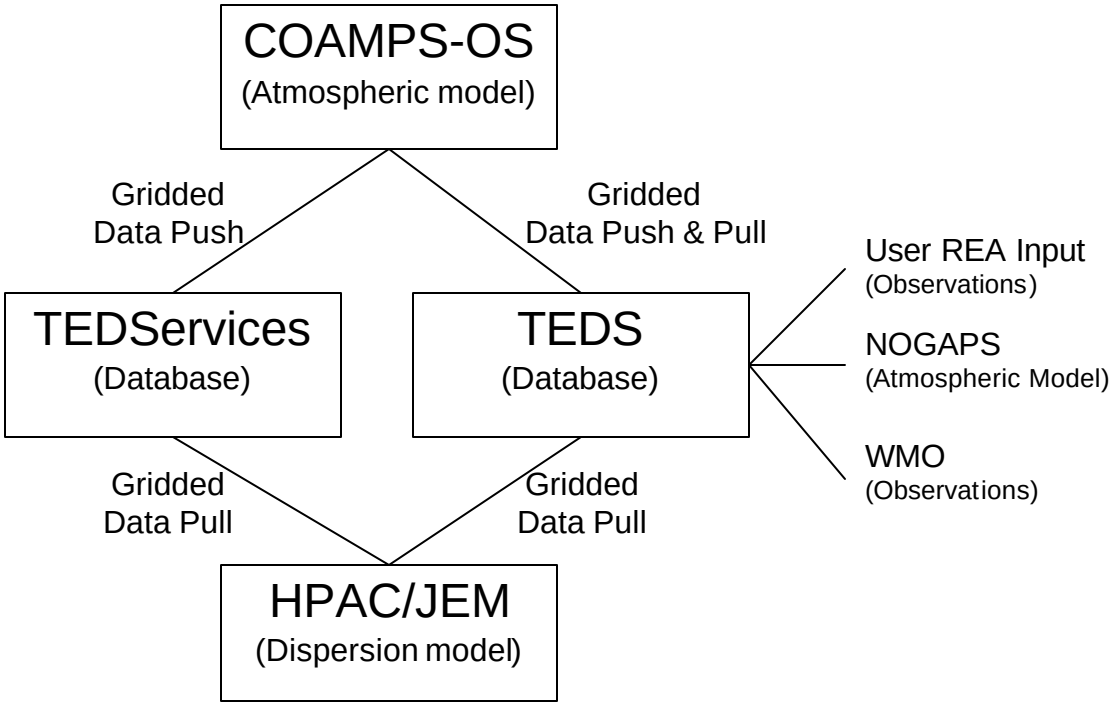

Figure 2. Thesis Data Flow

The thesis covers the inclusion of gridded atmospheric forecast data into TEDServices and the extraction of that same data for HPAC or JEM in the correct format. Data being inserted into both databases is in an Institute of Electrical and Electronics Engineers (IEEE) format, and

TEDServices accepts requests and returns the data as Java objects. HPAC or JEM must have the data MEDOC format. The IEEE to Java and Java to MEDOC formats are specific to METOC and WMD/WME applications. The HPAC or JEM application sends a request to the data servers, and waits for a formatted reply. The thesis software must query TEDServices, and it must determine which of the available models, available model resolutions, available valid times, and available parameters in the database should be sent back to the JEM or HPAC model. The software must then reformat the data from the Java objects to the required MEDOC format.

B. OPERATING PROCEDURES

This thesis project only addresses a small part of the complete JEM and HPAC projects. When completed, JEM will be used by all DOD branches for operational dispersion modeling. Since the end user of the application could be anyone in DOD, some assumptions must be made about the data needed from TEDServices. All observations, imagery and forecast data must be populated in TEDServices, but since the meteorological knowledge of the JEM user is unknown the VNE is used as the data source. This method assigns the requirements for the selection of the best available METOC data source to the DOD METOC command maintaining TEDServices and not the individual field user.

The primary reason for creating JEM is to reduce the number of WMD/WME applications in use by DOD personnel. The three primary models used in DOD are HPAC, Emergency Management Information System (D2PUFF), and the U.S. Navy’s

Chemical/Biological Agent Vapor, Liquid, and Solid Tracking model (VLSTRACK). The complexity and differing outputs from these applications have created problems for the DOD commanders who need this time-critical modeling output. JEM is designed to use the VNE to remove the METOC level of complexity. The new interface shown in Figure 3 has a single selection to import weather data, and leaves all of the METOC decisions the supporting METOC suppliers.

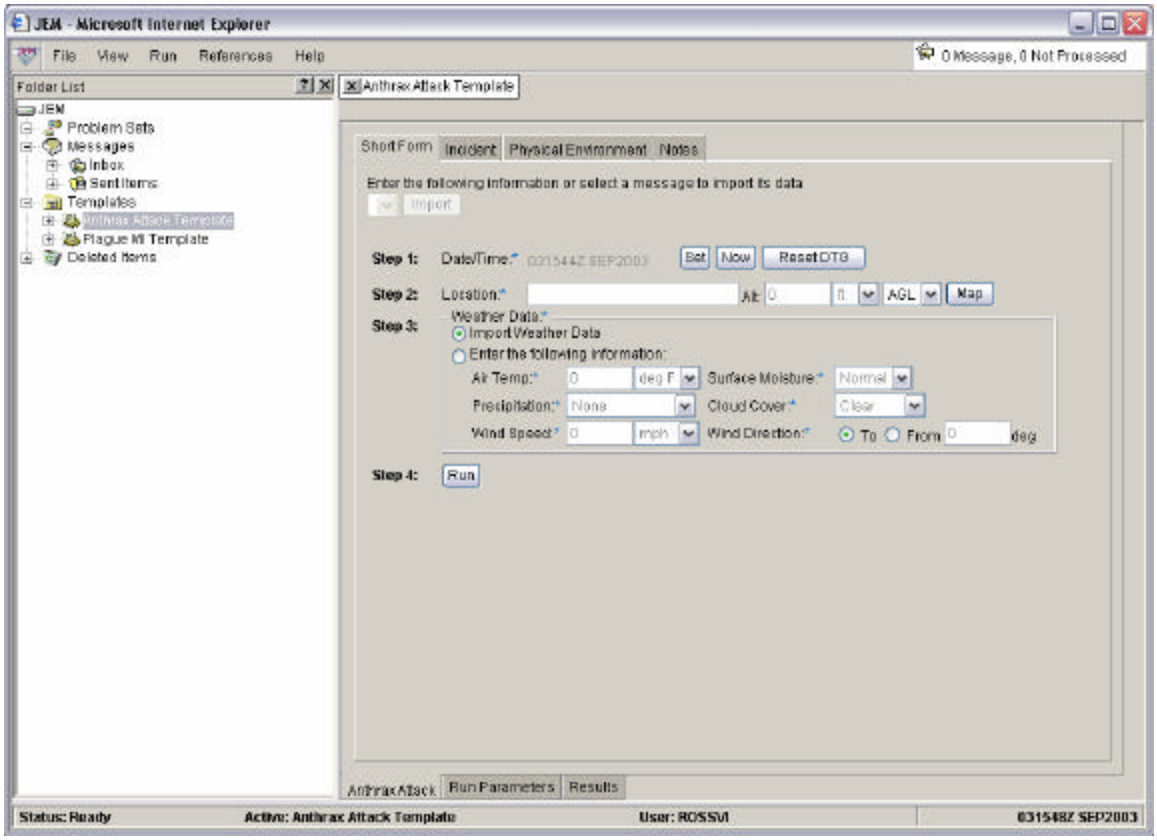

Figure 3. JEM Weather Selection Screen (From JEM)

One consideration not addressed by the JEM developers is the classification of the data in the project. Since WMD/WME planning and implementation is frequently classified, the system needs to be able to handle classified data, something TEDServices does not currently support. It assumes that all data within the database is

of the same classification as the network on which it resides. Unfortunately, if a user needs to provide the data to agencies with different classifications, it cannot currently be accomplished using TEDServices.

The management of classified data by JEM should make it much easier for the non-weather user to successfully use weather input for dispersion modeling. This will be accomplished not only by the software, but by the METOC professionals deciding which datasets should be included to create the VNE (Oceanographer of the Navy, 2002). This is extremely important in the operational theaters since different METOC data can create drastically different results from the dispersion model. By giving all users access to the best available baseline data and techniques for REA, the battle commander can be assured that modeling simulations will be consistent and the best available with current modeling techniques.

IV. PROGRAM DESCRIPTION

A. REQUIREMENTS

The JEMWeather program is written using Java SDK 1.4.1 (Schildt, 2002) for compatibility with the rest of the JEM code. Since it is Java, it is platform independent. As part of the larger JEM project, it will need constants defined in the primary JEM structure. For this thesis, the constants used are defined in the main procedure file written at NPS. Since the program has to interface with the TEDServices database structure, it must access the included TEDServices classes provided by the Naval Research Lab – Stennis Space Center (NRL-SSC).

B. INPUT / OUTPUT PARAMETERS

The thesis software tries to collect all required data from TEDServices. To accomplish this, two inputs are required from the user. The first is a bounding box of latitudes and longitudes passed as a float array with the format of “north, south, east, west”. Second is an incident time passed as a string with the format “YYYY.MM.DD HH MM SS.” The program retrieves the data from TEDServices and creates a data file of atmospheric variables at different atmospheric levels and times. It checks the database to determine which data should be given to the requesting routine. It checks to ensure that the requested properties of that data are available. The appropriate data is then formatted to the requested output format and written to a text output file. This is shown in Figure 4 with the sample input and output shown in Appendix A.

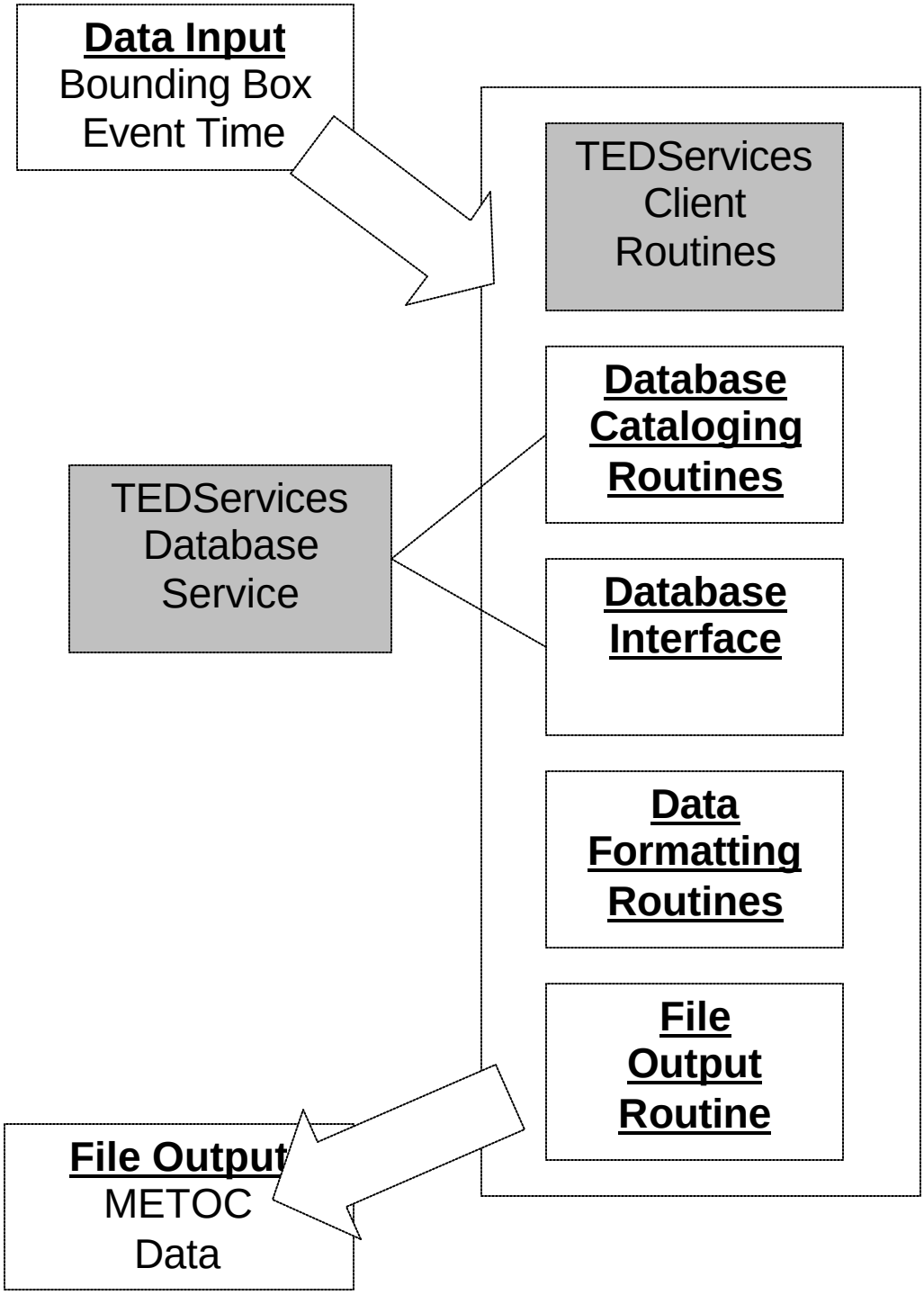

Figure 4. JEMWeather Procedure Flow

C. DATA STRUCTURE

The primary data structures used in this program are the inherent Java structures for hash maps and arrays, as well as some TEDServices-specific data structures. The most prominent TEDServices structure is the GridParameters3D object. This is a database-derived Java object that contains a complete volumetric representation of the field requested at all requested levels and times. The object also contains specific information identifying the structure and time of the associated data field. (Navy Research Laboratory – Stennis Space Center, 2003)

D. PROGRAM COMPONENTS

1. Program Constants

The JEMWeather program constants are listed in Table 1. Each of these is used in the thesis software, but will be stored in the JEM initialization file eventually. This initialization file will be changeable by the user through a JEM interface.

2. Primary Constructor

The primary Java constructor contains the initial call to the database to check the available data and valid times contained in TEDServices. This allows the user, during debugging, to determine which model fields have been returned by invoking the data printout routine. Primary error checking for the contents of the TEDServices as well as file systems is contained in this component. The final step within the constructor writes the data out to a file for use by JEM or HPAC. The constructor will pull all the fields specified in the 2D and 3D field variables. This

was left to be easily modified since the number and type of variables used by JEM should expand as the dispersion modelers make more use of the available meteorological fields.

VARIABLE	USAGE
database	Database address
Port	Port Contact Number
uName	Database login name
uPass	Database password
tauInc	Time offset for tau selection
htCoord	Height Coordinates
strAttributesCodes3	3D Grid Parameters
strAttributesCodes2	2D Grid Parameters
outputDir	Output location for the files
DUMMY_GRID	Name for place holding grid

Table 1. JEM Constants

3. Database Checker

The database checker does a full query of the available data in the TEDServices database. This component queries what models are available before checking the other criteria. If the database is functioning correctly the only model forecast data will be that which is considered by the METOC professionals to be the best available model. To select the VNE, the ATMOSPHERIC_FORECAST model type is the required argument. JEMWeather must also look for what forecast run times are available. This is the only way to

ensure that the most recent model run is used for the dispersion modeling. The selection should be the most recent run time, but it must ensure that the forecast extends far enough into the future to be useful in dispersion modeling. The final checks are of the model resolution, and to ensure the required data fields are in the database for the current forecast run time. If the database does not have the data it needs, it returns an error to the calling routine.

4. Data Printout

Data printout is designed for error checking. It shows the bounding box of the data retrieved from the database. It also shows the times, levels, and parameters retrieved from TEDServices. While this component is not used by the operational JEM user, it is very important for developers to be able to see what datasets and associated parameters are being returned to the primary data file written to disk.

5. Output File

An output file is created for JEM or HPAC. The output file has to meet the specified MEDOC standard created by DTRA. Each valid forecast time requested needs a specific header followed by data fields. Each of the entries in the file must be exactly spaced. No tabs or other special characters are allowed. Each section of the file is divided into three subsections.

The header subsection must have a 12-field format with the numbers right justified within the field. The date and

time fields must be the actual times that the data will be valid. This is accomplished using the Date function in Java. The valid time is calculated by adding the hours after forecast, taus, to the model run time. This header must also show the sigma level, terrain-following height above the ground, for each data point. This data is limited to the lower 20 atmospheric levels, and it must be listed with four numbers after the decimal.

The next subsection must contain the short names prescribed in the MEDOC format to show which fields will be included in the data section. These text names have to be in a 9-field formatting with the characters left justified. There are a number of 3D fields and a single 2D field that must be included. All the fields are listed in Table 2. (Defense Threat Reduction Agency, 2003) Since the database returns an array of gridded data, the grid for TOTAL_PRESSURE is requested as the PHI grid. This grid is used as a place holder that is then replaced by a computed grid based upon the SIGMA_HEIGHT and the TERRAIN_HEIGHT.

3D	**2D**
U_WIND W_WIND V_WIND PHI POTENTIAL_TEMPERATURE WATER_VAPOR_MIXING_RATIO	TERRAIN_HEIGHT

Table 2. Dispersion Parameters

The final subsection of each data section contains the actual data from the grids. These require the correct 12-field formatting and the four digits after the decimal place. The ordering for the data is specified in the

bottom of the header section described above. Each section is repeated for the full number of valid times collected from TEDServices.

E. ERROR CHECKING

This program uses built-in Java error checking that ensures that the files are present. Error checking in TEDServices is still immature, but the thesis program uses a generic try-catch routine to catch errors that occur while implementing TEDServices. The data can then be displayed to the user with a comment that the database is not functioning correctly. The most critical error checking is in the database-cataloging portion of the software. By ensuring that all the needed data is available, the program should prevent any incomplete calls to the database.

F. PROGRAM CONSTRAINTS AND MODIFICATIONS

The TEDServices database is still under development, and its interface has changed. One primary change is the ability to catalog TEDServices. This is essential for checking to see if there is data available. Without this addition, the database returned an error that the retrieval could not be accomplished. By using a separate interface a Web page can be retrieved and parsed to determine the current status of the database. A direct interface which allows the return of Java objects would be a more efficient.

Global longitude was stored in a 0° to 360° order starting at Greenwich, UK and continuing east around the

globe. Since most people enter longitude in a -180° to 180° arrangement, there was a conversion to accept values less than 0° and convert them. Recent updates have corrected this issue when using COAMPS, but will have to be resolved with different models as they are added to TEDServices.

The program also has to convert from model run time with tau to valid times to reduce the amount of data pulled. This is important to all limited-bandwidth users since it will not pull unneeded data from TEDServices. Since the input to the program is an event time for when the incident occurred, the program must convert it to a model run time and taus that follow the valid time. For this thesis, COAMPS-OS was set to run at 0000Z and 1200Z. An example is when the incident occurs at 2003 08 23 1130Z. Under the current TEDServices structure, the data that needs to be retrieved is model run time 2003 08 23 0000Z taus 10, 12, 14, 16, 18, 20, 22, and 24. An unrefined data pull would also pull the taus at 00, 02, 04, 06, and 08.

Each of these described parts of the program will accommodate the current level of the TEDServices interface, and will allow for easy changes if TEDServices changes. When TEDServices changes are complete the excess code should be removed to help reduce program size.

V. TEST AND EVALUATION

A. PERFORMANCE TESTING

Since this program is a component within the larger JEM program, it was designed to be small. Its performance will be affected by the available bandwidth since it has to collect data from TEDServices. In these experiments, TEDServices was run on the same network so network latency was minimal. To reduce necessary bandwidth, the program only requests data that is valid after the time of the WMD/WME event. This can easily reduce the total data requested from TEDServices by 25% to 50%. The timeliness of this new system allows for a test area with a final file size of approximately 2.5 MB to be downloaded, formatted and stored in approximately one minute. Because of the ability to do the check what is available in TEDServices, the program always returns the requested data or an error message.

B. PROGRAM / DATABASE EVALUATION

The initial program used the idea that the output file could be split with a regular expression such as a tab to create the formatting, but the HPAC and JEM programs could not use this output. This problem is made worse by the first column of each row being one space shorter than the other five columns in the row. After more testing, the use of the Java substring command seemed the best solution. The program has a string of spaces, and then selectively replaces the characters starting from the rightmost space to create the columns required for JEM and HPAC.

The JEM and HPAC dispersion modeling requires that the data files be in MEDOC format using a generic spherical projection. Since TEDServices cannot currently handle the mid-latitude native projection for COAMPS-OS, Lambert-Conformal Conic, the model was run using a spherical projection. Since the spherical projection does not have the best representation of the earth in the mid-latitudes, the spherical projection slows the model calculations and almost doubled the run time on the system used. A better alternative would be to modify TEDServices to accept gridded binary data in any projection type and reproject the data during extraction. TEDServices should be modified to accept a supplied variable that identifies the projection type of the data when stored.

Another needed parameter when selecting the grid is the security classification. Many models are run at different classifications, and some models can have multiple classifications. TEDServices must pull the data based on classification method as well. This would require rewriting part of the thesis code to allow for a new Grid3D parameter.

Although the VNE is stored in TEDServices, there is no way to specify model resolution. Since models are run at varying resolutions, cataloging and extraction should find and identify data of the proper resolution. The code for this thesis works because only one model resolution is submitted to TEDServices in a given time frame; if there are multiple resolutions over the same area, the program will retrieve them all. This presents a problem to the JEM or HPAC code since neither can deal with multiple sets of

model data. Fixing this will not be as easy since the resolution is usually chosen by a trained meteorologist since there are many factors to consider. Another approach would be to allow the VNE to hold only the best-resolution model for the area in concern. This would make the process easier in many cases, but would not help when multiple models cover an area with the same resolution. It also ignores how close the requested area is to the boundary of a specific nest.

The current program pulls all available height levels, but it needs a more specific ability to pull one of the three available height coordinates. The current coordinate systems are MILLIBAR, SIGMA, and DISTANCE. With JEM or HPAC, the coordinate system used is the SIGMA level, the height above the terrain. A way to resolve this problem within this thesis is to only put the SIGMA level data into TEDServices.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. SUMMARY AND RECOMMENDATIONS

A. SUMMARY

This program is one of the new meteorology interfaces being designed for the JEM system, and it streamlines the data interface required for dispersion modeling. The HPAC interface requires individual usernames and passwords, knowledge of available models, and which database servers are available. This thesis interface collects all of this information from configuration files making it easier for the end user.

It also attempts to retrieve the correct data without retrieving all available data and wasting bandwidth. The program creates an output file in the prescribed MEDOC format for backward compatibility with HPAC. It outputs a file instead of just passing the variables to allow for a single download of meteorological data for multiple possible dispersion model runs.

There are issues that need to be solved to make this program more robust. The most important is the classification issue, which will require a change to the database, but should be done immediately. There can also be some future removal of code as the database becomes better equipped to deal with valid times instead of forecast times and TEDServices starts storing grids in the more standard convention of -180° and 180° of longitude.

Specific testing of timing within the database was shown to be superior in two different facets. Since the database is run locally, there are minimal network latency issues. This was shown repeatedly when attempting to

collect the required data for the dispersion model. Times from the remote database ranged from 6 to 40 minutes, while times from TEDServices ranged from 30 seconds to 3 minutes. This is probably also attributed to the fact that TEDServices does not interpolate points from the model. The current TEDS database will give interpolated values based upon the data request. TEDServices does not interpolate values, but returns the actual values stored in the database. Each of these factors is of benefit when attempting to do REA and WMD/WME scenarios.

B. RECOMMENDATIONS

Recommendations to make this program and the overall TEDServices database more useful to the METOC community would include adding the following functionalities.

1. The ability to ingest and output data in World Meteorological Organization Gridded Binary (GRIB) format. This is the standard for gridded model data and should be fully supported so that dispersion modelers can use any model produced.

2. TEDServices needs to support other METOC models. Many nonmilitary models available are not supported.

3. The ability to output IEEE file formats for use by other models. This will allow TEDServices to be used as a data source for atmospheric modeling.

4. The ability to ingest and output atmospheric and oceanic observations. Many times the individual observations help with the initialization of the dispersion model, but this data is not currently available from TEDServices.

5. The ability to set a security classification level for individual models or observations. Military data has different classification levels, and the available data needs to be marked and handled appropriately.

6. The ability to select grid size. There are times when a smaller grid size is not preferable.

7. The ability to select data based on the forecast time and the valid time. Currently the database stores items with the forecast time and time offsets from it, but many applications look for data starting with the time that the application needs. Without this unneeded data will be retrieved.

The overall data flow of the REA and dispersion modeling needs to be streamlined as shown in Figure 5.

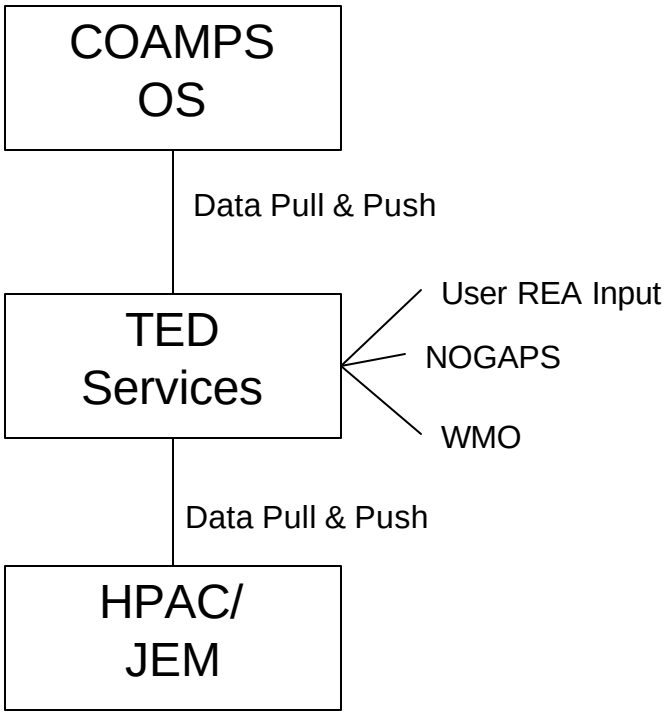

Figure 5. Proposed Data Flow

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A: TEST RUN

A. SAMPLE COMMAND LINE CALL

JEMWeather 33.0 32.5 -117.0 -117.5 2003.11.05.13.00

B. SAMPLE SINGLE TAU OUTPUT

```
FFFFFFFF

NRLCOAMPS

          5          11          03          13           0           0

          5          11          03          13           0           0

          8           8          20           0           6           1

          0           0           0           0           0           0

          0           0           0

     10.0000     30.0000     55.0000     90.0000    140.0000    215.0000

    330.0000    500.0000    750.0000   1100.0000   1600.0000   2300.0000

   3100.0000   3900.0000   4800.0000   5800.0000   6800.0000   7800.0000

   8675.0000   9425.0000      0.0900      0.0900 -999999.0000 -999999.0000

     32.4590   -117.5680      0.0000      0.0000      0.0000      0.0000

      0.0000

U        V        W        T        H        PHI

M/S      M/S      M/S      KELVIN   GM/GM    METERS

TOPO     METERS

      1.6951      1.3273      0.9199      0.3220     -0.7150     -1.8808

     -2.6554     -2.8663      1.5813      1.2433      0.8369      0.1841

     -0.9546     -2.1005     -2.7175     -2.8325      1.4557      1.1561

      0.7441      0.0114     -1.1138     -2.1845     -2.6800     -2.6646

      1.3045      1.0469      0.6042     -0.2698     -1.2978     -2.0116

     -2.4041     -2.4618      1.1201      0.9163      0.4470     -0.5539

     -1.5781     -2.1020     -2.3470     -2.3121      0.9111      0.7817

      0.3050     -0.7743     -1.8389     -2.2899     -2.4509     -2.3574
```

 0.6986 0.6466 0.1285 -1.0293 -1.9939 -2.3060
-2.3905 -2.2693 0.4895 0.3887 -0.3533 -1.4543
-2.0329 -2.1648 -2.2299 -2.1294 1.7110 1.3393
 0.9323 0.3584 -0.7347 -2.1939 -3.2932 -3.5711
 1.5957 1.2546 0.8510 0.2262 -0.9724 -2.4151
-3.3046 -3.3869 1.4687 1.1672 0.7627 0.0438
-1.1323 -2.3750 -3.1115 -3.1074 1.3151 1.0584
 0.6381 -0.3008 -1.6721 -2.7565 -3.1508 -2.8795
 1.1280 0.9283 0.4856 -0.6059 -2.0311 -2.9157
-3.0873 -2.6934 0.9160 0.7940 0.3425 -0.8102
-2.1803 -2.8753 -2.9385 -2.5651 0.7005 0.6649
 0.1686 -1.0708 -2.2725 -2.7277 -2.7084 -2.3822
 0.4921 0.4314 -0.3381 -1.6472 -2.4252 -2.5701
-2.5295 -2.2732 1.7198 1.3467 0.9410 0.4065
-0.4702 -1.4435 -2.0209 -2.0499 1.6039 1.2618
 0.8621 0.2861 -0.6717 -1.6140 -1.9933 -1.8291
 1.4760 1.1746 0.7814 0.1094 -0.8322 -1.6102
-1.8319 -1.4947 1.3206 1.0657 0.6813 -0.1109
-1.1049 -1.6870 -1.6712 -1.1683 1.1317 0.9366
 0.5453 -0.2955 -1.1954 -1.5614 -1.4360 -0.9313
 0.9177 0.8044 0.4087 -0.4030 -1.1365 -1.3335
-1.1722 -0.7418 0.6995 0.6857 0.2854 -0.4857
-1.0061 -1.0823 -0.9247 -0.5664 0.4914 0.4948
 0.0991 -0.5261 -0.8212 -0.8488 -0.7881 -0.5337
 1.7272 1.3539 0.9513 0.4797 0.0266 -0.2089
-0.2307 -0.0767 1.6107 1.2692 0.8768 0.3861
-0.0941 -0.3129 -0.2590 -0.0193 1.4821 1.1820
 0.8122 0.3063 -0.1327 -0.3010 -0.2066 0.1119
 1.3251 1.0730 0.7481 0.3351 -0.0098 -0.1184
 0.0007 0.3345 1.1342 0.9474 0.6604 0.3038

0.0898 0.0903 0.2136 0.5589 0.9179 0.8229

0.5869 0.3038 0.1929 0.2486 0.3729 0.7168

0.6965 0.7244 0.7006 0.5260 0.3762 0.3946

0.5234 0.9039 0.4905 0.5905 0.6425 0.6035

0.5776 0.5761 0.6630 1.0342 1.7346 1.3632

0.9704 0.5800 0.4517 0.6745 0.9218 1.0735

1.6178 1.2792 0.9038 0.5267 0.3863 0.5975

0.8950 1.1174 1.4886 1.1926 0.8643 0.6019

0.5428 0.6833 0.9499 1.2242 1.3303 1.0856

0.8348 0.7160 0.8085 1.0154 1.2202 1.4192

1.1374 0.9679 0.8309 0.8461 1.0698 1.3466

1.4917 1.6418 0.9183 0.8584 0.9141 1.0899

1.3434 1.6124 1.7116 1.8286 0.6953 0.7713

1.0231 1.3154 1.5808 1.8152 1.8932 2.0276

0.4933 0.6871 1.0291 1.4019 1.7236 1.8938

1.9371 2.0904 1.7436 1.3782 1.0114 0.7258

0.7787 1.1375 1.3947 1.4484 1.6266 1.2963

0.9578 0.7134 0.7452 1.0485 1.3348 1.4495

1.4974 1.2132 0.9399 0.8245 0.9428 1.1770

1.4128 1.5576 1.3383 1.1111 0.9337 0.9540

1.1898 1.4766 1.6562 1.7279 1.1446 1.0025

0.9650 1.1122 1.4467 1.7972 1.9123 1.9085

0.9243 0.9020 1.0523 1.3344 1.6779 2.0118

2.0854 2.0507 0.7011 0.8234 1.1526 1.4892

1.7737 2.0315 2.1067 2.1574 0.5049 0.7564

1.1347 1.4924 1.8204 2.0159 2.0652 2.1734

1.7563 1.4062 1.1031 0.9886 1.2138 1.6517

1.9691 2.0924 1.6398 1.3298 1.0699 0.9989

1.1488 1.4539 1.7568 1.9268 1.5124 1.2561

1.0663 1.0615 1.2142 1.4208 1.6613 1.8568

1.3548 1.1631 1.0649 1.1218 1.3015 1.5385

1.7360 1.8678 1.1621 1.0595 1.0853 1.2148

1.4514 1.7542 1.8975 1.9420 0.9413 0.9559

1.1372 1.3741 1.6427 1.9630 2.0632 2.0319

0.7193 0.8751 1.2022 1.4766 1.7369 2.0267

2.1079 2.1027 0.5289 0.8111 1.1476 1.4213

1.7559 2.0115 2.0734 2.1165 1.7911 1.4917

1.3501 1.4591 1.7381 2.0504 2.2325 2.2674

1.6768 1.4304 1.3475 1.4693 1.6185 1.7695

1.9172 1.9762 1.5586 1.3805 1.3543 1.4519

1.5063 1.5216 1.6001 1.6912 1.4097 1.3037

1.3201 1.3454 1.3006 1.2962 1.3467 1.4242

1.2213 1.1828 1.2166 1.1583 1.0732 1.1127

1.1517 1.2212 0.9974 1.0365 1.0966 0.9929

0.8748 0.9488 1.0127 1.0916 0.7740 0.8977

0.9658 0.7854 0.6540 0.7671 0.8859 1.0452

0.5793 0.7415 0.7078 0.5175 0.5305 0.6921

0.8479 1.0860 2.4901 2.6982 2.7762 2.5691

2.2285 1.9945 1.8537 1.7625 2.4605 2.6538

2.6841 2.4603 2.0759 1.7995 1.6913 1.6235

2.4162 2.5839 2.5681 2.3120 1.9177 1.6023

1.4836 1.4686 2.3166 2.4452 2.3802 2.0659

1.6538 1.3836 1.2933 1.3337 2.1251 2.2043

2.0904 1.7216 1.3363 1.1656 1.1373 1.2752

1.8465 1.8728 1.7227 1.3500 1.0168 0.9614

1.0468 1.3024 1.4967 1.4828 1.3134 0.9465

0.7163 0.8066 1.0286 1.4217 1.1285 1.0897

0.8615 0.5903 0.5942 0.8011 1.1049 1.5952

4.0980 4.0699 3.8242 3.3953 2.9253 2.5244

2.1583 1.8667 3.7571 3.6325 3.3506 2.9687

2.5490	2.1959	1.9253	1.7352	3.3171	3.1213
2.8308	2.4928	2.1436	1.8509	1.6814	1.6587
2.8022	2.5797	2.3175	2.0211	1.7342	1.5411
1.4900	1.6512	2.2921	2.0907	1.8796	1.6237
1.4094	1.3236	1.3857	1.7338	1.8822	1.7298
1.5724	1.3631	1.2119	1.2270	1.4095	1.9116
1.6236	1.5240	1.4046	1.2204	1.1444	1.2751
1.5817	2.2036	1.5127	1.4458	1.3119	1.1976
1.2644	1.4732	1.8563	2.5515	1.7088	1.5493
1.4599	1.3801	1.2882	1.2027	1.2090	1.4603
1.6482	1.6042	1.5746	1.5077	1.4211	1.3970
1.5203	1.8928	1.6715	1.6871	1.6621	1.5768
1.4940	1.5399	1.8172	2.3676	1.7359	1.7587
1.7027	1.5948	1.5460	1.6764	2.0908	2.8335
1.7988	1.7924	1.6944	1.5835	1.6038	1.8323
2.3671	3.2780	1.8358	1.7864	1.6608	1.5786
1.6957	2.0468	2.6901	3.7077	1.8435	1.7610
1.6490	1.6463	1.8923	2.3798	3.1222	4.1751
1.8360	1.7407	1.6895	1.8349	2.2146	2.7745
3.5681	4.6200	2.7152	2.6151	2.4385	2.3423
2.5089	2.9851	3.7771	4.8534	2.9120	2.8049
2.6357	2.5422	2.7516	3.3550	4.2778	5.4324
3.0558	2.9445	2.7978	2.7540	3.0172	3.7001
4.7262	5.9442	3.1652	3.0629	2.9564	3.0067
3.3844	4.1113	5.1452	6.3951	3.2513	3.1768
3.1262	3.2853	3.7740	4.5175	5.5287	6.7872
3.3194	3.2886	3.2974	3.5353	4.1079	4.8863
5.8825	7.1264	3.3790	3.3988	3.4788	3.8235
4.4878	5.3128	6.3060	7.4880	3.4422	3.5123
3.7271	4.2326	4.9331	5.7327	6.7070	7.8391

6.5295	6.6572	6.7558	6.9495	7.3761	8.0220
8.8437	9.7667	6.5822	6.7442	6.8972	7.1120
7.5525	8.2623	9.1426	10.0874	6.6238	6.8240
7.0304	7.3005	7.7598	8.4814	9.3919	10.3460
6.6728	6.9050	7.1473	7.4859	8.0245	8.7418
9.6122	10.5675	6.7365	6.9883	7.2477	7.6446
8.2510	8.9583	9.8010	10.7701	6.8175	7.0769
7.3375	7.7528	8.4034	9.1370	9.9850	10.9734
6.9255	7.1828	7.4471	7.9178	8.6287	9.4019
10.2704	11.2424	7.0689	7.3217	7.6691	8.2534
8.9593	9.7098	10.5947	11.5723	10.2205	10.4050
10.5704	10.7935	11.1649	11.6567	12.2302	12.8151
10.2637	10.4757	10.6729	10.8894	11.2476	11.7788
12.4027	13.0309	10.3056	10.5410	10.7603	10.9960
11.3468	11.8797	12.5403	13.2065	10.3376	10.5859
10.8118	11.0818	11.4881	12.0172	12.6617	13.3657
10.3578	10.6052	10.8248	11.1321	11.5984	12.1341
12.7853	13.5379	10.3736	10.6084	10.8150	11.1382
11.6619	12.2502	12.9405	13.7427	10.4051	10.6204
10.8212	11.2044	11.8060	12.4606	13.2021	14.0157
10.4798	10.6805	10.9552	11.4492	12.0625	12.7294
13.5187	14.3600	12.7571	12.9016	13.0687	13.3167
13.7035	14.1704	14.6648	15.1571	12.7602	12.9127
13.0856	13.3107	13.6876	14.2156	14.7980	15.3799
12.7801	12.9321	13.1003	13.3271	13.6988	14.2499
14.9034	15.5613	12.7945	12.9422	13.1006	13.3500
13.7742	14.3348	14.9996	15.7207	12.7920	12.9321
13.0811	13.3662	13.8517	14.4306	15.1197	15.8964
12.7793	12.9097	13.0567	13.3731	13.9269	14.5646
15.2936	16.1060	12.7845	12.9080	13.0728	13.4697

14.1093	14.8070	15.5699	16.3688	12.8553	12.9862
13.2436	13.7566	14.4095	15.1152	15.9103	16.7125
16.4943	16.5727	16.7143	16.9529	17.3225	17.7519
18.1885	18.6186	16.4454	16.5199	16.6645	16.8891
17.2652	17.7711	18.3053	18.8244	16.4158	16.4793
16.6175	16.8507	17.2372	17.7824	18.3941	18.9838
16.3937	16.4472	16.5756	16.8338	17.2754	17.8348
18.4567	19.0970	16.3705	16.4170	16.5361	16.8240
17.3168	17.8853	18.5194	19.1984	16.3500	16.3897
16.5040	16.8116	17.3551	17.9640	18.6206	19.3168
16.3496	16.3834	16.5100	16.8852	17.4977	18.1454
18.8144	19.4784	16.4049	16.4449	16.6561	17.1372
17.7544	18.3965	19.0810	19.7286	18.9152	19.0219
19.2233	19.5086	19.8701	20.2274	20.5370	20.8002
18.9067	19.0130	19.2263	19.5112	19.8835	20.3007
20.6790	20.9995	18.8992	18.9950	19.2036	19.4966
19.8795	20.3313	20.7716	21.1459	18.8957	18.9773
19.1692	19.4704	19.8910	20.3564	20.8081	21.2252
18.8959	18.9627	19.1322	19.4377	19.8846	20.3483
20.8069	21.2597	18.9018	18.9523	19.0977	19.3920
19.8574	20.3393	20.8132	21.2875	18.9194	18.9486
19.0738	19.3956	19.9013	20.4042	20.8882	21.3450
18.9643	18.9724	19.1393	19.5368	20.0406	20.5343
21.0327	21.4763	19.6663	19.8000	19.9946	20.2342
20.4911	20.7006	20.8335	20.9040	19.7425	19.8905
20.1086	20.3667	20.6423	20.8866	21.0520	21.1471
19.8072	19.9583	20.1856	20.4590	20.7504	21.0226
21.2290	21.3638	19.8743	20.0192	20.2410	20.5175
20.8297	21.1261	21.3623	21.5430	19.9543	20.0864
20.2918	20.5669	20.8973	21.2068	21.4660	21.6921

20.0515 20.1687 20.3575 20.6248 20.9682 21.2942

21.5760 21.8394 20.1694 20.2747 20.4479 20.7251

21.0975 21.4426 21.7447 22.0221 20.3201 20.4165

20.6074 20.9322 21.3178 21.6668 21.9876 22.2595

19.7065 19.7610 19.8478 19.9676 20.1049 20.2246

20.3073 20.3566 19.8070 19.8862 19.9992 20.1402

20.2912 20.4243 20.5184 20.5755 19.9093 20.0036

20.1381 20.2971 20.4608 20.6113 20.7270 20.8080

20.0254 20.1229 20.2683 20.4370 20.6114 20.7867

20.9294 21.0422 20.1649 20.2563 20.4009 20.5775

20.7723 20.9674 21.1269 21.2747 20.3294 20.4144

20.5599 20.7445 20.9558 21.1630 21.3363 21.5197

20.5196 20.6122 20.7614 20.9510 21.1858 21.4072

21.5961 21.8035 20.7498 20.8610 21.0220 21.2400

21.4983 21.7273 21.9379 22.1492 21.0621 20.9936

20.9327 20.9298 21.0185 21.1860 21.4049 21.6456

21.1009 21.0578 21.0138 21.0114 21.0998 21.2851

21.5294 21.7894 21.1561 21.1287 21.1078 21.1179

21.2118 21.4176 21.6892 21.9727 21.2480 21.2207

21.2161 21.2521 21.3646 21.5869 21.8704 22.1768

21.3829 21.3447 21.3427 21.4012 21.5442 21.7814

22.0675 22.4030 21.5513 21.5009 21.4971 21.5726

21.7496 22.0074 22.3026 22.6712 21.7336 21.6870

21.6992 21.8045 22.0290 22.3052 22.5977 22.9693

21.9309 21.9168 21.9760 22.1380 22.3909 22.6602

22.9607 23.3215

-1.9067 -1.5434 -1.3544 -1.2630 -0.9991 -0.7143

-0.6575 -0.8992 -1.7737 -1.3942 -1.1592 -0.9488

-0.5216 -0.2785 -0.4085 -0.8113 -1.6596 -1.2804

-1.0197 -0.7518 -0.4136 -0.3119 -0.4367 -0.8925

-1.5870 -1.2209 -0.9499 -0.6513 -0.4376 -0.4314

-0.5702 -1.0537 -1.5432 -1.2026 -0.9329 -0.6271

-0.4441 -0.4980 -0.7349 -1.2284 -1.4758 -1.1729

-0.9309 -0.7053 -0.6144 -0.7720 -1.1121 -1.5608

-1.3262 -1.0711 -0.9340 -0.9290 -0.9641 -1.1221

-1.4777 -1.8821 -1.0768 -0.9151 -0.9550 -1.0403

-1.0928 -1.2691 -1.5730 -1.9666 -1.9221 -1.5533

-1.3613 -1.2802 -1.0577 -1.0653 -1.6684 -2.3485

-1.7877 -1.4026 -1.1655 -0.9706 -0.5592 -0.5888

-1.2933 -2.1143 -1.6722 -1.2879 -1.0277 -0.7740

-0.4352 -0.4928 -1.2221 -2.1238 -1.5984 -1.2279

-0.9608 -0.6774 -0.5741 -0.7995 -1.4174 -2.2732

-1.5532 -1.2094 -0.9450 -0.6647 -0.7621 -1.1505

-1.7091 -2.4670 -1.4846 -1.1800 -0.9409 -0.7559

-1.0355 -1.5518 -2.1577 -2.8203 -1.3341 -1.0765

-0.9435 -1.0211 -1.4563 -1.9584 -2.5606 -3.1684

-1.0815 -0.9098 -0.9862 -1.3220 -1.7379 -2.1907

-2.7229 -3.3267 -1.9291 -1.5573 -1.3635 -1.2979

-1.3188 -1.8188 -2.5944 -3.3045 -1.7942 -1.4060

-1.1678 -0.9991 -0.9890 -1.4151 -2.1906 -2.9903

-1.6781 -1.2911 -1.0325 -0.8141 -0.8669 -1.2946

-2.0417 -2.9015 -1.6035 -1.2308 -0.9714 -0.7975

-0.9656 -1.4265 -2.0995 -2.9624 -1.5575 -1.2125

-0.9609 -0.8763 -1.1050 -1.5779 -2.2196 -3.0638

-1.4882 -1.1839 -0.9577 -0.9603 -1.2708 -1.7884

-2.4712 -3.2854 -1.3377 -1.0797 -0.9632 -1.1495

-1.5435 -2.0537 -2.7645 -3.5612 -1.0836 -0.8987

-0.9877 -1.3383 -1.7529 -2.2810 -2.9744 -3.7803

-1.9330 -1.5591 -1.3658 -1.3211 -1.4666 -2.0120

-2.7555 -3.4805 -1.7980 -1.4075 -1.1723 -1.0369

-1.1473 -1.6039 -2.3429 -3.1402 -1.6814 -1.2926

-1.0412 -0.8722 -0.9990 -1.4262 -2.1389 -2.9919

-1.6062 -1.2331 -0.9864 -0.8588 -1.0348 -1.4653

-2.1217 -2.9909 -1.5594 -1.2162 -0.9824 -0.9135

-1.1085 -1.5372 -2.1793 -3.0415 -1.4898 -1.1890

-0.9808 -0.9533 -1.1961 -1.6709 -2.3671 -3.2064

-1.3400 -1.0833 -0.9530 -1.0534 -1.3726 -1.8824

-2.6280 -3.4409 -1.0853 -0.8690 -0.8420 -1.1060

-1.5216 -2.1149 -2.8564 -3.6532 -1.9345 -1.5593

-1.3700 -1.3513 -1.5580 -2.0744 -2.7591 -3.4604

-1.7996 -1.4079 -1.1812 -1.0851 -1.2249 -1.6588

-2.3426 -3.1190 -1.6827 -1.2936 -1.0555 -0.9420

-1.0726 -1.4559 -2.1056 -2.9447 -1.6070 -1.2356

-1.0076 -0.9250 -1.0806 -1.4469 -2.0436 -2.9147

-1.5595 -1.2211 -1.0139 -0.9524 -1.1026 -1.4600

-2.0595 -2.9457 -1.4897 -1.1957 -1.0091 -0.9506

-1.1246 -1.5382 -2.2233 -3.1097 -1.3413 -1.0863

-0.9303 -0.9667 -1.2248 -1.7201 -2.5050 -3.3792

-1.0860 -0.8370 -0.7076 -0.8916 -1.3048 -1.9699

-2.8068 -3.6717 -1.9322 -1.5577 -1.3784 -1.3980

-1.6901 -2.2745 -2.9800 -3.6466 -1.7983 -1.4073

-1.1978 -1.1568 -1.3793 -1.8721 -2.5722 -3.3204

-1.6811 -1.2944 -1.0787 -1.0249 -1.2217 -1.6338

-2.2828 -3.1096 -1.6053 -1.2390 -1.0353 -1.0004

-1.1959 -1.5636 -2.1494 -3.0377 -1.5571 -1.2275

-1.0440 -1.0086 -1.1715 -1.5214 -2.1262 -3.0706

-1.4873 -1.2038 -1.0323 -0.9839 -1.1638 -1.5854

-2.3134 -3.2953 -1.3403 -1.0898 -0.9348 -0.9823

-1.2553 -1.7772 -2.6392 -3.6339 -1.0847 -0.8224

-0.6934 -0.8838 -1.3161 -2.0324 -2.9764 -3.9472

-1.9225 -1.5536 -1.3994 -1.5141 -1.9476 -2.6222

-3.3176 -3.8767 -1.7912 -1.4056 -1.2365 -1.3453

-1.7513 -2.3553 -3.0411 -3.6556 -1.6737 -1.2958

-1.1302 -1.2484 -1.6407 -2.1646 -2.7940 -3.4804

-1.5979 -1.2448 -1.0901 -1.2063 -1.5774 -2.0452

-2.6148 -3.3823 -1.5497 -1.2372 -1.0937 -1.1664

-1.4589 -1.8838 -2.4786 -3.3388 -1.4804 -1.2163

-1.0763 -1.0962 -1.3488 -1.8035 -2.5222 -3.4501

-1.3355 -1.1007 -0.9756 -1.0736 -1.3724 -1.8839

-2.7309 -3.6755 -1.0836 -0.8236 -0.7214 -0.9432

-1.3755 -2.0706 -2.9911 -3.9015 -1.8886 -1.5476

-1.4789 -1.7797 -2.2622 -2.8819 -3.4429 -3.8016

-1.7651 -1.4103 -1.3708 -1.7203 -2.2264 -2.8069

-3.3620 -3.7426 -1.6452 -1.3129 -1.3202 -1.6892

-2.2288 -2.7775 -3.3009 -3.7275 -1.5662 -1.2752

-1.3342 -1.7066 -2.2792 -2.8371 -3.3382 -3.8206

-1.5181 -1.2757 -1.3759 -1.7472 -2.3232 -2.9119

-3.4563 -3.9884 -1.4537 -1.2649 -1.4040 -1.8046

-2.4378 -3.0993 -3.7263 -4.2526 -1.3209 -1.1930

-1.4076 -1.9714 -2.6988 -3.3918 -4.0680 -4.5418

-1.0995 -1.0399 -1.3387 -2.0486 -2.8655 -3.6526

-4.3265 -4.7351 -1.6812 -1.7247 -1.9501 -2.2947

-2.6610 -3.0781 -3.3991 -3.5280 -1.6678 -1.8040

-2.0862 -2.4283 -2.7555 -3.1156 -3.4598 -3.6640

-1.6633 -1.8626 -2.1667 -2.4930 -2.8292 -3.1656

-3.5237 -3.8245 -1.6837 -1.9342 -2.2529 -2.5764

-2.9584 -3.3237 -3.6895 -4.0631 -1.7527 -2.0432

-2.3758 -2.7102 -3.1325 -3.5525 -3.9565 -4.3501

-1.8862 -2.2068 -2.5525 -2.9221 -3.4229 -3.9114

-4.3437 -4.6684 -2.0913 -2.4303 -2.8104 -3.3203

-3.8915 -4.3695 -4.7636 -4.9613 -2.3456 -2.6830

-3.0998 -3.7222 -4.3207 -4.7892 -5.0746 -5.1372

-2.4518 -2.7785 -3.1725 -3.5424 -3.8092 -3.9889

-4.0192 -3.9159 -3.0905 -3.4115 -3.7464 -4.0165

-4.1789 -4.2820 -4.3260 -4.2986 -3.6725 -3.9453

-4.1844 -4.3402 -4.4248 -4.4564 -4.4887 -4.5189

-4.1570 -4.3527 -4.4875 -4.5463 -4.5844 -4.5760

-4.5746 -4.6131 -4.4895 -4.6040 -4.6568 -4.6557

-4.6657 -4.6436 -4.6141 -4.6044 -4.6407 -4.6977

-4.7066 -4.6878 -4.7042 -4.6800 -4.6122 -4.5069

-4.6174 -4.6526 -4.6701 -4.7174 -4.7387 -4.6811

-4.5614 -4.3606 -4.4443 -4.4771 -4.5332 -4.6659

-4.7121 -4.6467 -4.4570 -4.1887 -3.9116 -3.9269

-3.9276 -3.8854 -3.7505 -3.5418 -3.2980 -3.1092

-3.8054 -3.8336 -3.8786 -3.9148 -3.8737 -3.7270

-3.5130 -3.3183 -3.7419 -3.7926 -3.8774 -3.9649

-3.9693 -3.8398 -3.6309 -3.4100 -3.7335 -3.8203

-3.9448 -4.0585 -4.0596 -3.9166 -3.6967 -3.4336

-3.7920 -3.9196 -4.0746 -4.1892 -4.1580 -3.9898

-3.7415 -3.4223 -3.9137 -4.0712 -4.2299 -4.3102

-4.2221 -4.0096 -3.7135 -3.3489 -4.0727 -4.2373

-4.3675 -4.3761 -4.1993 -3.9416 -3.6226 -3.2580

-4.2167 -4.3504 -4.4180 -4.3585 -4.1428 -3.8662

-3.5443 -3.2204 -1.8149 -1.2004 -0.6436 -0.2573

-0.1302 -0.2192 -0.4535 -0.7922 -1.8968 -1.2965

-0.7505 -0.3686 -0.2166 -0.2615 -0.4335 -0.7111

-2.1081 -1.5355 -1.0228 -0.6700 -0.4925 -0.4769

-0.5979 -0.8231 -2.4181 -1.8903 -1.4265 -1.1103

-0.9153 -0.8437 -0.9286 -1.1196 -2.7703 -2.3004

-1.8947 -1.6164 -1.4265 -1.3346 -1.3836 -1.5440

-3.1142	-2.7097	-2.3705	-2.1323	-1.9494	-1.8472
-1.8529	-1.9825	-3.4326	-3.1010	-2.8325	-2.6006
-2.4015	-2.3012	-2.2916	-2.4022	-3.7426	-3.4818
-3.2555	-3.0294	-2.8454	-2.7351	-2.7160	-2.8131
-3.0433	-2.9290	-2.8171	-2.7332	-2.7867	-2.9345
-3.1430	-3.4241	-3.0355	-2.9272	-2.8199	-2.7377
-2.7571	-2.9013	-3.1207	-3.4080	-3.1280	-3.0321
-2.9501	-2.9166	-2.9487	-3.0859	-3.3174	-3.5991
-3.2933	-3.2141	-3.1639	-3.1890	-3.2682	-3.4081
-3.6414	-3.9101	-3.5014	-3.4418	-3.4234	-3.4914
-3.6233	-3.7915	-4.0120	-4.2557	-3.7195	-3.6820
-3.7001	-3.8015	-3.9574	-4.1421	-4.3354	-4.5445
-3.9271	-3.9184	-3.9815	-4.0771	-4.2164	-4.4050
-4.5830	-4.7589	-4.1306	-4.1559	-4.2306	-4.3087
-4.4383	-4.5858	-4.7356	-4.8805	-3.7715	-3.7321
-3.6898	-3.6579	-3.7200	-3.8318	-3.9681	-4.1321
-3.5935	-3.5354	-3.4805	-3.4415	-3.4735	-3.5861
-3.7358	-3.9117	-3.5550	-3.4929	-3.4422	-3.4276
-3.4475	-3.5291	-3.6707	-3.8334	-3.6076	-3.5514
-3.5135	-3.5301	-3.5598	-3.6085	-3.7277	-3.8668
-3.7125	-3.6642	-3.6397	-3.6759	-3.7277	-3.7777
-3.8690	-3.9796	-3.8526	-3.8094	-3.8034	-3.8594
-3.9209	-3.9781	-4.0380	-4.1205	-4.0320	-3.9985
-4.0274	-4.0727	-4.1191	-4.1912	-4.2455	-4.3137
-4.2786	-4.2699	-4.3133	-4.3435	-4.3960	-4.4551
-4.5123	-4.5825	-5.4712	-5.4341	-5.3948	-5.3796
-5.4673	-5.5918	-5.7156	-5.8381	-5.2108	-5.1581
-5.1109	-5.0930	-5.1685	-5.3283	-5.5047	-5.6749
-5.0971	-5.0343	-4.9792	-4.9725	-5.0326	-5.1777
-5.3818	-5.5887	-5.0763	-5.0113	-4.9531	-4.9648

-5.0270	-5.1388	-5.3437	-5.5769	-5.0897	-5.0260
-4.9699	-4.9956	-5.0764	-5.1846	-5.3759	-5.6207
-5.1062	-5.0410	-4.9968	-5.0457	-5.1440	-5.2637
-5.4348	-5.6733	-5.1298	-5.0621	-5.0545	-5.1115
-5.2097	-5.3527	-5.5220	-5.7436	-5.2046	-5.1475
-5.1641	-5.2239	-5.3380	-5.4762	-5.6423	-5.8433
-7.5278	-7.5324	-7.5339	-7.5172	-7.5317	-7.5305
-7.4815	-7.3944	-7.3561	-7.3445	-7.3385	-7.3289
-7.3435	-7.3689	-7.3585	-7.3088	-7.2353	-7.2028
-7.1782	-7.1736	-7.1802	-7.2016	-7.2238	-7.2160
-7.1515	-7.1002	-7.0567	-7.0521	-7.0557	-7.0627
-7.1052	-7.1335	-7.0791	-7.0150	-6.9618	-6.9542
-6.9656	-6.9828	-7.0329	-7.0794	-7.0071	-6.9357
-6.8899	-6.8977	-6.9182	-6.9486	-6.9879	-7.0314
-6.9461	-6.8746	-6.8643	-6.8787	-6.8948	-6.9412
-6.9754	-7.0077	-6.9306	-6.8749	-6.8889	-6.9027
-6.9357	-6.9777	-7.0063	-7.0263	-7.0070	-6.9862
-6.9849	-6.9557	-6.9220	-6.8611	-6.7527	-6.6102
-6.9359	-6.9059	-6.9058	-6.9003	-6.8846	-6.8464
-6.7628	-6.6398	-6.8219	-6.7739	-6.7635	-6.7742
-6.7681	-6.7391	-6.6876	-6.5971	-6.6949	-6.6248
-6.5981	-6.6080	-6.6041	-6.5805	-6.5607	-6.5028
-6.5721	-6.4817	-6.4424	-6.4414	-6.4395	-6.4370
-6.4366	-6.3952	-6.4651	-6.3606	-6.3227	-6.3254
-6.3206	-6.3270	-6.3193	-6.2773	-6.3868	-6.2809
-6.2683	-6.2615	-6.2371	-6.2463	-6.2297	-6.1849
-6.3586	-6.2705	-6.2705	-6.2437	-6.2247	-6.2260
-6.2030	-6.1560	-6.8704	-6.7971	-6.7422	-6.6681
-6.5738	-6.4487	-6.2766	-6.0578	-6.8690	-6.7956
-6.7482	-6.6980	-6.6314	-6.5315	-6.3812	-6.1710

-6.7520 -6.6704 -6.6246 -6.5949 -6.5516 -6.4781

-6.3606 -6.1724 -6.5743 -6.4757 -6.4243 -6.4030

-6.3728 -6.3216 -6.2386 -6.0732 -6.3888 -6.2701

-6.2103 -6.1858 -6.1607 -6.1309 -6.0716 -5.9187

-6.2313 -6.0954 -6.0302 -6.0059 -5.9794 -5.9566

-5.8982 -5.7462 -6.1176 -5.9714 -5.9133 -5.8796

-5.8368 -5.8096 -5.7373 -5.5788 -6.0510 -5.9075

-5.8531 -5.7996 -5.7506 -5.7132 -5.6244 -5.4595

-6.3616 -6.2193 -6.1093 -5.9660 -5.7415 -5.4569

-5.0984 -4.6665 -6.4799 -6.3443 -6.2400 -6.1246

-5.9369 -5.6572 -5.2903 -4.8326 -6.4155 -6.2857

-6.1918 -6.0944 -5.9400 -5.6908 -5.3328 -4.8596

-6.2353 -6.1031 -6.0190 -5.9251 -5.7812 -5.5681

-5.2338 -4.7520 -6.0218 -5.8814 -5.8026 -5.7013

-5.5473 -5.3576 -5.0507 -4.5663 -5.8374 -5.6884

-5.6042 -5.4911 -5.3256 -5.1408 -4.8470 -4.3629

-5.7074 -5.5497 -5.4493 -5.3167 -5.1345 -4.9335

-4.6227 -4.1380 -5.6218 -5.4549 -5.3334 -5.1664

-4.9629 -4.7407 -4.4012 -3.9172 -4.4215 -4.2195

-4.0915 -3.9051 -3.5708 -3.1526 -2.6554 -2.1130

-4.7017 -4.5083 -4.3784 -4.2261 -3.9210 -3.4610

-2.9052 -2.2975 -4.7482 -4.5709 -4.4539 -4.3157

-4.0417 -3.5893 -2.9986 -2.3326 -4.6245 -4.4591

-4.3640 -4.2148 -3.9394 -3.5250 -2.9375 -2.2308

-4.4371 -4.2758 -4.2006 -4.0281 -3.7218 -3.3409

-2.7848 -2.0617 -4.2745 -4.1140 -4.0398 -3.8365

-3.5014 -3.1317 -2.6032 -1.8815 -4.1789 -4.0173

-3.9081 -3.6597 -3.2989 -2.9092 -2.3770 -1.6728

-4.1368 -3.9664 -3.7964 -3.4787 -3.0889 -2.6766

-2.1273 -1.4522

0.0000	0.0000	0.0000	-0.0003	-0.0051	-0.0219
-0.0405	-0.0496	0.0000	0.0000	0.0000	-0.0005
-0.0105	-0.0302	-0.0463	-0.0531	0.0000	0.0000
0.0000	-0.0009	-0.0111	-0.0346	-0.0519	-0.0549
0.0000	0.0000	0.0000	-0.0027	-0.0155	-0.0319
-0.0471	-0.0562	0.0000	0.0000	0.0000	-0.0063
-0.0200	-0.0311	-0.0465	-0.0577	0.0000	0.0000
0.0000	-0.0083	-0.0269	-0.0372	-0.0553	-0.0680
0.0000	0.0000	0.0000	-0.0189	-0.0402	-0.0472
-0.0635	-0.0701	0.0000	0.0000	-0.0046	-0.0284
-0.0401	-0.0463	-0.0620	-0.0661	0.0009	0.0010
0.0008	0.0011	-0.0043	-0.0226	-0.0442	-0.0556
0.0008	0.0009	0.0010	0.0012	-0.0091	-0.0306
-0.0499	-0.0582	0.0007	0.0009	0.0013	0.0010
-0.0097	-0.0343	-0.0551	-0.0594	0.0006	0.0009
0.0015	-0.0015	-0.0156	-0.0357	-0.0524	-0.0610
0.0004	0.0008	0.0016	-0.0049	-0.0212	-0.0354
-0.0519	-0.0624	-0.0001	0.0004	0.0016	-0.0068
-0.0275	-0.0408	-0.0599	-0.0721	-0.0008	-0.0002
0.0016	-0.0173	-0.0418	-0.0512	-0.0687	-0.0747
-0.0013	0.0001	-0.0032	-0.0286	-0.0439	-0.0510
-0.0677	-0.0713	0.0017	0.0018	0.0016	0.0023
-0.0016	-0.0199	-0.0442	-0.0577	0.0014	0.0017
0.0020	0.0028	-0.0054	-0.0275	-0.0491	-0.0581
0.0013	0.0017	0.0024	0.0027	-0.0060	-0.0302
-0.0527	-0.0576	0.0012	0.0018	0.0029	0.0014
-0.0135	-0.0364	-0.0525	-0.0584	0.0008	0.0015
0.0031	-0.0011	-0.0192	-0.0361	-0.0511	-0.0587
-0.0001	0.0007	0.0031	-0.0023	-0.0238	-0.0393
-0.0568	-0.0661	-0.0015	-0.0004	0.0030	-0.0120

-0.0378	-0.0485	-0.0655	-0.0695	-0.0025	0.0001
0.0006	-0.0234	-0.0414	-0.0490	-0.0649	-0.0678
0.0023	0.0025	0.0022	0.0032	0.0038	-0.0093
-0.0291	-0.0407	0.0020	0.0024	0.0027	0.0040
0.0011	-0.0146	-0.0317	-0.0387	0.0018	0.0024
0.0034	0.0044	0.0001	-0.0166	-0.0329	-0.0361
0.0017	0.0024	0.0040	0.0059	-0.0053	-0.0231
-0.0324	-0.0344	0.0012	0.0021	0.0043	0.0048
-0.0082	-0.0214	-0.0291	-0.0313	-0.0002	0.0009
0.0044	0.0042	-0.0098	-0.0218	-0.0322	-0.0347
-0.0021	-0.0008	0.0053	-0.0026	-0.0189	-0.0275
-0.0395	-0.0392	-0.0036	-0.0002	0.0056	-0.0080
-0.0216	-0.0285	-0.0388	-0.0392	0.0032	0.0035
0.0030	0.0044	0.0101	0.0056	-0.0045	-0.0115
0.0027	0.0033	0.0036	0.0055	0.0083	0.0040
-0.0039	-0.0079	0.0025	0.0033	0.0045	0.0075
0.0075	0.0033	-0.0018	-0.0042	0.0024	0.0033
0.0053	0.0105	0.0072	0.0005	0.0005	-0.0004
0.0016	0.0028	0.0059	0.0109	0.0085	0.0033
0.0053	0.0060	-0.0002	0.0011	0.0062	0.0111
0.0107	0.0054	0.0043	0.0064	-0.0029	-0.0014
0.0069	0.0089	0.0090	0.0043	-0.0017	0.0011
-0.0050	-0.0014	0.0091	0.0118	0.0064	0.0023
-0.0011	0.0009	0.0045	0.0048	0.0040	0.0056
0.0130	0.0145	0.0120	0.0079	0.0038	0.0046
0.0049	0.0070	0.0123	0.0160	0.0151	0.0127
0.0035	0.0046	0.0059	0.0100	0.0129	0.0186
0.0207	0.0178	0.0034	0.0045	0.0068	0.0130
0.0161	0.0184	0.0249	0.0230	0.0023	0.0036
0.0071	0.0143	0.0201	0.0215	0.0311	0.0310

-0.0003	0.0012	0.0063	0.0150	0.0256	0.0262
0.0329	0.0344	-0.0040	-0.0027	0.0050	0.0163
0.0290	0.0291	0.0277	0.0279	-0.0071	-0.0039
0.0081	0.0227	0.0245	0.0259	0.0268	0.0265
0.0064	0.0068	0.0054	0.0065	0.0120	0.0163
0.0188	0.0169	0.0054	0.0065	0.0065	0.0080
0.0130	0.0194	0.0223	0.0215	0.0051	0.0064
0.0075	0.0105	0.0146	0.0257	0.0303	0.0278
0.0049	0.0062	0.0081	0.0132	0.0185	0.0248
0.0357	0.0336	0.0033	0.0047	0.0074	0.0144
0.0223	0.0279	0.0437	0.0423	-0.0004	0.0008
0.0040	0.0147	0.0299	0.0347	0.0486	0.0474
-0.0058	-0.0053	-0.0000	0.0177	0.0351	0.0393
0.0414	0.0382	-0.0105	-0.0084	0.0027	0.0225
0.0283	0.0359	0.0388	0.0350	0.0091	0.0095
0.0069	0.0058	0.0082	0.0139	0.0208	0.0223
0.0077	0.0089	0.0081	0.0071	0.0120	0.0176
0.0224	0.0242	0.0073	0.0087	0.0089	0.0092
0.0133	0.0258	0.0292	0.0287	0.0071	0.0083
0.0088	0.0118	0.0152	0.0200	0.0331	0.0319
0.0046	0.0058	0.0062	0.0123	0.0149	0.0204
0.0407	0.0385	-0.0008	-0.0003	-0.0006	0.0108
0.0219	0.0273	0.0474	0.0437	-0.0085	-0.0102
-0.0079	0.0137	0.0243	0.0313	0.0357	0.0313
-0.0158	-0.0181	-0.0067	0.0102	0.0139	0.0294
0.0327	0.0282	0.0137	0.0134	0.0061	0.0032
0.0045	0.0101	0.0196	0.0234	0.0113	0.0120
0.0071	0.0038	0.0125	0.0145	0.0189	0.0224
0.0103	0.0112	0.0082	0.0074	0.0140	0.0259
0.0241	0.0251	0.0097	0.0102	0.0083	0.0130

0.0143	0.0137	0.0262	0.0247	0.0059	0.0063
0.0050	0.0145	0.0084	0.0100	0.0320	0.0277
-0.0021	-0.0027	-0.0035	0.0124	0.0136	0.0142
0.0378	0.0311	-0.0138	-0.0168	-0.0119	0.0144
0.0072	0.0132	0.0177	0.0148	-0.0251	-0.0278
-0.0110	-0.0049	-0.0115	0.0105	0.0128	0.0116
0.0117	0.0132	0.0089	0.0149	0.0162	0.0153
0.0217	0.0239	0.0082	0.0116	0.0092	0.0115
0.0260	0.0179	0.0181	0.0219	0.0070	0.0112
0.0112	0.0162	0.0264	0.0345	0.0238	0.0255
0.0075	0.0113	0.0132	0.0263	0.0257	0.0180
0.0280	0.0246	0.0049	0.0086	0.0119	0.0313
0.0157	0.0121	0.0342	0.0256	-0.0022	0.0002
0.0047	0.0310	0.0217	0.0148	0.0402	0.0286
-0.0141	-0.0141	-0.0018	0.0341	0.0071	0.0093
0.0119	0.0096	-0.0268	-0.0248	0.0017	-0.0035
-0.0231	0.0039	0.0040	0.0050	0.0210	0.0312
0.0360	0.0510	0.0476	0.0382	0.0410	0.0408
0.0193	0.0295	0.0325	0.0388	0.0538	0.0335
0.0308	0.0347	0.0195	0.0284	0.0314	0.0386
0.0478	0.0509	0.0325	0.0347	0.0196	0.0261
0.0301	0.0476	0.0425	0.0288	0.0365	0.0299
0.0145	0.0196	0.0256	0.0519	0.0269	0.0207
0.0414	0.0263	0.0036	0.0078	0.0154	0.0508
0.0338	0.0221	0.0478	0.0286	-0.0121	-0.0099
0.0080	0.0557	0.0157	0.0157	0.0164	0.0096
-0.0291	-0.0238	0.0126	0.0037	-0.0223	0.0106
0.0076	0.0047	0.0359	0.0480	0.0567	0.0761
0.0696	0.0627	0.0669	0.0620	0.0313	0.0440
0.0503	0.0604	0.0748	0.0498	0.0490	0.0495

0.0303	0.0426	0.0492	0.0586	0.0669	0.0654
0.0425	0.0422	0.0302	0.0409	0.0487	0.0681
0.0599	0.0412	0.0422	0.0306	0.0248	0.0342
0.0438	0.0714	0.0393	0.0287	0.0412	0.0195
0.0128	0.0213	0.0306	0.0659	0.0418	0.0244
0.0452	0.0191	-0.0052	0.0001	0.0180	0.0697
0.0209	0.0165	0.0146	0.0024	-0.0265	-0.0208
0.0164	0.0062	-0.0221	0.0126	0.0067	-0.0005
0.0444	0.0608	0.0723	0.0891	0.0747	0.0632
0.0595	0.0448	0.0430	0.0615	0.0726	0.0806
0.0844	0.0528	0.0471	0.0390	0.0469	0.0668
0.0790	0.0854	0.0838	0.0716	0.0448	0.0382
0.0510	0.0705	0.0832	0.0978	0.0833	0.0566
0.0477	0.0322	0.0472	0.0655	0.0779	0.0999
0.0643	0.0458	0.0466	0.0223	0.0341	0.0503
0.0600	0.0873	0.0608	0.0377	0.0486	0.0201
0.0136	0.0245	0.0387	0.0856	0.0378	0.0277
0.0213	0.0048	-0.0112	-0.0045	0.0279	0.0210
-0.0048	0.0236	0.0144	0.0027	0.0393	0.0640
0.0785	0.0886	0.0676	0.0453	0.0297	0.0094
0.0471	0.0727	0.0877	0.0910	0.0856	0.0514
0.0373	0.0230	0.0587	0.0846	0.1002	0.1033
0.0945	0.0774	0.0503	0.0381	0.0668	0.0917
0.1065	0.1167	0.1015	0.0740	0.0602	0.0429
0.0625	0.0856	0.0991	0.1168	0.0877	0.0667
0.0601	0.0365	0.0450	0.0660	0.0771	0.0989
0.0786	0.0552	0.0561	0.0292	0.0195	0.0358
0.0493	0.0896	0.0538	0.0404	0.0286	0.0099
-0.0080	0.0025	0.0324	0.0336	0.0144	0.0302
0.0164	0.0006	0.0123	0.0405	0.0565	0.0619

0.0417	0.0148	-0.0066	-0.0232	0.0297	0.0580
0.0759	0.0792	0.0720	0.0418	0.0225	0.0083
0.0478	0.0762	0.0955	0.1013	0.0933	0.0762
0.0506	0.0364	0.0593	0.0871	0.1060	0.1172
0.1069	0.0835	0.0661	0.0488	0.0562	0.0825
0.1001	0.1166	0.0982	0.0788	0.0651	0.0441
0.0384	0.0631	0.0790	0.0972	0.0842	0.0634
0.0538	0.0318	0.0130	0.0347	0.0513	0.0815
0.0586	0.0438	0.0274	0.0111	-0.0119	0.0047
0.0324	0.0382	0.0262	0.0277	0.0105	-0.0044
-0.0348	-0.0071	0.0089	0.0088	-0.0077	-0.0322
-0.0522	-0.0590	-0.0018	0.0266	0.0454	0.0471
0.0347	0.0110	-0.0089	-0.0180	0.0284	0.0584
0.0797	0.0852	0.0728	0.0517	0.0304	0.0187
0.0460	0.0767	0.0987	0.1062	0.0939	0.0717
0.0507	0.0370	0.0450	0.0747	0.0956	0.1036
0.0908	0.0714	0.0488	0.0348	0.0279	0.0556
0.0748	0.0820	0.0708	0.0532	0.0308	0.0184
0.0047	0.0298	0.0479	0.0563	0.0441	0.0296
0.0095	0.0007	-0.0138	0.0074	0.0271	0.0293
0.0251	0.0111	-0.0073	-0.0159	-0.0768	-0.0546
-0.0413	-0.0484	-0.0593	-0.0764	-0.0899	-0.0862
-0.0298	-0.0073	0.0074	0.0038	-0.0151	-0.0284
-0.0430	-0.0443	0.0130	0.0381	0.0548	0.0538
0.0340	0.0113	0.0002	-0.0055	0.0370	0.0641
0.0823	0.0788	0.0608	0.0420	0.0220	0.0159
0.0372	0.0640	0.0817	0.0743	0.0645	0.0468
0.0186	0.0152	0.0192	0.0436	0.0604	0.0509
0.0395	0.0267	-0.0062	-0.0052	-0.0033	0.0187
0.0331	0.0138	0.0121	0.0002	-0.0213	-0.0189

-0.0161	0.0031	0.0085	0.0026	0.0073	-0.0207
-0.0373	-0.0359	-0.1018	-0.0910	-0.0823	-0.0961
-0.1022	-0.1099	-0.1152	-0.1026	-0.0503	-0.0407
-0.0337	-0.0419	-0.0662	-0.0673	-0.0728	-0.0658
-0.0006	0.0113	0.0184	0.0100	-0.0147	-0.0358
-0.0338	-0.0312	0.0282	0.0427	0.0518	0.0353
0.0130	0.0021	-0.0133	-0.0105	0.0301	0.0452
0.0556	0.0308	0.0241	0.0125	-0.0182	-0.0115
0.0126	0.0260	0.0381	0.0108	-0.0014	-0.0065
-0.0473	-0.0345	-0.0079	0.0051	0.0145	-0.0325
-0.0249	-0.0323	-0.0535	-0.0399	-0.0146	-0.0019
-0.0140	-0.0295	-0.0145	-0.0521	-0.0659	-0.0528
-0.1201	-0.1213	-0.1170	-0.1365	-0.1387	-0.1355
-0.1309	-0.1104	-0.0743	-0.0782	-0.0787	-0.0888
-0.1161	-0.1046	-0.0982	-0.0825	-0.0246	-0.0274
-0.0299	-0.0429	-0.0688	-0.0858	-0.0700	-0.0580
0.0069	0.0062	0.0061	-0.0202	-0.0453	-0.0468
-0.0554	-0.0447	0.0120	0.0124	0.0154	-0.0248
-0.0298	-0.0324	-0.0637	-0.0503	-0.0025	-0.0028
0.0042	-0.0386	-0.0534	-0.0492	-0.0962	-0.0763
-0.0192	-0.0177	-0.0120	-0.0842	-0.0721	-0.0731
-0.0950	-0.0731	-0.0201	-0.0147	-0.0431	-0.0704
-0.0459	-0.0878	-0.1006	-0.0756	-0.1274	-0.1348
-0.1332	-0.1561	-0.1559	-0.1450	-0.1318	-0.1051
-0.0939	-0.1046	-0.1091	-0.1198	-0.1471	-0.1265
-0.1095	-0.0866	-0.0505	-0.0614	-0.0693	-0.0840
-0.1086	-0.1199	-0.0932	-0.0736	-0.0198	-0.0295
-0.0355	-0.0658	-0.0919	-0.0862	-0.0876	-0.0710
-0.0121	-0.0210	-0.0235	-0.0716	-0.0767	-0.0723
-0.1009	-0.0832	-0.0226	-0.0323	-0.0308	-0.0817

-0.0984	-0.0869	-0.1349	-0.1117	-0.0359	-0.0436
-0.0415	-0.1272	-0.1139	-0.1085	-0.1314	-0.1051
-0.0344	-0.0354	-0.0726	-0.1078	-0.0779	-0.1192
-0.1319	-0.0977				
287.2728	287.2589	287.2138	287.0438	286.0132	283.5964
281.3989	279.9792	287.2470	287.2227	287.1655	286.9795
286.0800	283.5970	281.3848	280.0706	287.2233	287.1880
287.1051	286.8608	285.9863	284.2845	281.6049	280.1737
287.2043	287.1713	287.0498	286.4640	284.5992	282.6804
281.2366	280.2232	287.1854	287.1581	286.9984	286.1567
283.7022	281.8720	281.0241	280.2501	287.1639	287.1405
286.9547	286.0580	283.5481	281.7505	281.0675	280.2942
287.1499	287.1145	286.8392	285.8498	283.1684	281.5661
280.7975	280.0440	287.1438	287.0605	286.4337	284.2901
282.1889	281.2556	280.5356	279.7443	287.0520	287.0391
286.9918	286.8164	285.7902	284.1635	282.7889	281.7920
287.0262	287.0031	286.9437	286.7542	285.8471	284.1174
282.7283	281.8136	287.0024	286.9684	286.8848	286.6431
285.7130	284.1610	282.7536	281.8808	286.9834	286.9500
286.8279	286.2534	285.0213	283.6841	282.6950	281.9023
286.9651	286.9359	286.7774	285.9722	284.5060	283.3994
282.7259	281.9116	286.9448	286.9186	286.7327	285.8705
284.4377	283.4143	282.9122	282.0360	286.9310	286.8929
286.6174	285.7173	284.2110	283.3161	282.6545	281.8289
286.9231	286.8377	286.2259	284.9734	283.7170	283.1469
282.4055	281.5393	286.7951	286.7821	286.7317	286.5571
285.7495	284.7042	283.7663	282.8886	286.7691	286.7460
286.6833	286.4966	285.8549	284.7406	283.7284	282.8198
286.7448	286.7110	286.6255	286.3953	285.7830	284.8727
283.7462	282.8058	286.7255	286.6903	286.5725	286.2216

285.4949 284.5750 283.6950 282.7597 286.7074 286.6747
286.5277 286.1271 285.1881 284.3464 283.6490 282.7340
286.6876 286.6576 286.4840 286.0880 285.1425 284.2675
283.6816 282.7688 286.6735 286.6325 286.3840 286.0739
284.9499 284.0836 283.3869 282.5655 286.6627 286.5836
286.3165 285.5753 284.5255 283.8875 283.1549 282.3693
286.4451 286.4316 286.3782 286.2070 285.5201 284.6220
283.7441 282.8555 286.4189 286.3954 286.3297 286.1492
285.6627 284.6825 283.7036 282.7545 286.3940 286.3600
286.2735 286.0688 285.6254 284.8019 283.6974 282.6967
286.3742 286.3372 286.2262 286.0127 285.4340 284.5589
283.6255 282.5966 286.3559 286.3207 286.1881 285.9660
285.1773 284.3420 283.5638 282.5378 286.3366 286.3043
286.1512 285.9377 285.1263 284.2454 283.5736 282.5552
286.3218 286.2815 286.1267 285.9402 284.9003 284.0291
283.2563 282.3476 286.3090 286.2416 286.1154 285.4703
284.4408 283.7976 283.0030 282.1635 285.9513 285.9371
285.8811 285.7140 285.0638 284.1964 283.3042 282.3831
285.9247 285.9005 285.8326 285.6594 285.2153 284.2575
283.2494 282.2581 285.8990 285.8645 285.7789 285.6092
285.2059 284.3914 283.2399 282.1936 285.8784 285.8401
285.7364 285.5768 285.0481 284.1640 283.1694 282.0912
285.8600 285.8232 285.7066 285.5486 284.8118 283.9594
283.1129 282.0308 285.8411 285.8082 285.6917 285.5318
284.7576 283.8521 283.1166 282.0539 285.8260 285.7881
285.6751 285.5244 284.4987 283.5987 282.7848 281.8709
285.8124 285.7558 285.6577 285.0320 283.9924 283.3321
282.5403 281.7494 285.2166 285.2006 285.1430 284.9818
284.3712 283.5779 282.7902 282.0110 285.1894 285.1636
285.0950 284.9287 284.5013 283.5848 282.6495 281.7820

285.1627 285.1273 285.0440 284.8826 284.4760 283.6673
282.5509 281.6078 285.1409 285.1020 285.0054 284.8498
284.3123 283.4124 282.4261 281.4328 285.1225 285.0853
284.9818 284.8286 284.0852 283.2102 282.3555 281.3428
285.1039 285.0718 284.9708 284.8190 284.0416 283.1157
282.3735 281.3730 285.0890 285.0548 284.9551 284.8108
283.7794 282.8630 282.0594 281.2239 285.0756 285.0286
284.9359 284.3112 283.2611 282.5919 281.8340 281.1457
284.0964 284.0774 284.0202 283.8839 283.4276 282.8451
282.2745 281.6974 284.0684 284.0406 283.9742 283.8334
283.5255 282.7996 282.0945 281.4622 284.0391 284.0048
283.9275 283.7914 283.4659 282.8094 281.9192 281.2299
284.0146 283.9776 283.8879 283.7543 283.2787 282.4934
281.6862 280.9391 283.9967 283.9609 283.8676 283.7302
283.0221 282.2002 281.4471 280.6575 283.9792 283.9495
283.8611 283.7156 282.9356 281.9933 281.2710 280.4736
283.9653 283.9380 283.8516 283.7031 282.6483 281.6886
280.8833 280.2398 283.9535 283.9157 283.8339 283.2034
282.1230 281.4005 280.6618 280.1874 282.4554 282.4230
282.3967 282.4886 282.3475 282.0933 281.8322 281.5298
282.4330 282.3874 282.3547 282.4196 282.4003 282.0215
281.6881 281.4046 282.4214 282.3538 282.3192 282.3679
282.3238 282.0474 281.6166 281.3668 282.4068 282.3245
282.2917 282.3463 282.1899 281.8551 281.5883 281.3592
282.3708 282.3080 282.2946 282.3729 282.0472 281.7479
281.5996 281.3702 282.3218 282.3010 282.3287 282.4432
282.1051 281.7398 281.6467 281.3900 282.3111 282.3125
282.3797 282.5423 281.9767 281.5965 281.3918 281.2002
282.3085 282.3246 282.4125 282.1325 281.5385 281.3378
281.1108 280.9650 280.2753 280.3700 280.6176 280.9932

281.1754 281.2702 281.3133 281.1782 280.3416 280.4589
280.6963 281.0086 281.2474 281.1768 281.1075 280.9437
280.4301 280.5558 280.7714 281.0190 281.1684 281.1290
280.9185 280.7534 280.5236 280.6373 280.8095 281.0140
281.0225 280.8798 280.7709 280.6022 280.6138 280.7099
280.8362 281.0106 280.8425 280.7181 280.7056 280.5391
280.7195 280.7901 280.8694 281.0209 280.8475 280.7021
280.7791 280.5962 280.8310 280.8713 280.9271 281.1319
280.8114 280.7109 280.7026 280.5462 280.9362 280.9537
281.0478 280.9340 280.6507 280.7225 280.6576 280.4824
280.8885 281.0051 281.0232 280.9575 280.6648 280.3865
280.2209 280.0687 281.1284 281.2124 281.1945 281.0811
280.8848 280.5566 280.4077 280.2923 281.2790 281.3219
281.2701 281.1533 280.9827 280.7937 280.5940 280.4791
281.3440 281.3448 281.2665 281.1959 281.0406 280.8576
280.7560 280.5681 281.3416 281.3130 281.2297 281.2251
281.0188 280.9028 280.8523 280.5698 281.2960 281.2598
281.1884 281.2309 281.0722 280.9388 280.9167 280.5257
281.2339 281.2040 281.1649 281.3129 281.0293 280.8996
280.7255 280.2800 281.1688 281.1502 281.2008 281.0999
280.8264 280.7895 280.4860 279.9732 280.4453 280.4643
280.5479 280.7683 280.9071 280.9854 280.9188 280.5799
280.3423 280.3806 280.4957 280.7124 280.9482 280.9432
280.8221 280.4463 280.2140 280.2764 280.4178 280.6514
280.8621 280.9021 280.6454 280.2222 280.0941 280.1779
280.3302 280.5793 280.7093 280.6357 280.4049 279.8922
279.9924 280.0824 280.2226 280.4724 280.4442 280.3345
280.1096 279.5077 279.8981 279.9785 280.0898 280.3057
280.2266 280.0182 279.7884 279.1002 279.7993 279.8541
279.9359 280.1678 279.8719 279.5983 279.1941 278.4962

279.6856 279.6951 279.7914 279.6993 279.3357 279.1174

278.5974 277.8901 279.2324 279.2766 279.2999 279.3021

279.0057 278.4961 277.8524 277.0508 279.2757 279.3148

279.3270 279.3003 279.1172 278.5251 277.8100 276.9618

279.2824 279.3062 279.2968 279.2496 279.0371 278.5375

277.7218 276.8452 279.2483 279.2536 279.2213 279.1661

278.8568 278.2824 277.5774 276.6514 279.1731 279.1599

279.1092 279.0604 278.6042 278.0457 277.4123 276.4308

279.0637 279.0370 278.9751 278.9231 278.4519 277.8322

277.2300 276.1930 278.9420 278.9041 278.8360 278.8114

278.1421 277.4980 276.7519 275.7503 278.8251 278.7682

278.7267 278.3691 277.6749 277.1495 276.3307 275.3533

274.4537 274.4479 274.4474 274.4503 274.1554 273.6899

273.1364 272.4523 274.4191 274.4057 274.4057 274.4139

274.2732 273.7188 273.0680 272.3164 274.3554 274.3360

274.3366 274.3551 274.2218 273.7707 272.9906 272.1775

274.2738 274.2531 274.2558 274.2815 274.0482 273.5230

272.8483 271.9623 274.1766 274.1584 274.1638 274.1964

273.7912 273.2746 272.6750 271.7160 274.0646 274.0503

274.0594 274.0928 273.6540 273.0542 272.4752 271.4371

273.9489 273.9385 273.9476 273.9911 273.3130 272.6556

271.9033 270.8769 273.8434 273.8315 273.8469 273.4773

272.7222 272.1621 271.3084 270.2847 270.0281 270.0565

270.0931 270.1331 269.8662 269.4224 268.8925 268.2185

270.0130 270.0450 270.0894 270.1351 270.0148 269.4615

268.8074 268.0433 269.9843 270.0219 270.0764 270.1336

270.0110 269.5424 268.7336 267.8839 269.9442 269.9856

270.0479 270.1150 269.8822 269.3250 268.6093 267.6677

269.8948 269.9359 270.0005 270.0735 269.6626 269.1087

268.4625 267.4346 269.8384 269.8761 269.9380 270.0076

269.5569	268.9155	268.2853	267.1749	269.7823	269.8163
269.8694	269.9335	269.2302	268.5320	267.7317	266.6463
269.7313	269.7593	269.8065	269.4339	268.6507	268.0647
267.1821	266.1245	264.5211	264.5804	264.6499	264.7172
264.4515	263.9852	263.4094	262.6575	264.4585	264.5159
264.5818	264.6431	264.5201	263.9353	263.2288	262.3899
264.4089	264.4646	264.5286	264.5861	264.4505	263.9449
263.0760	262.1512	264.3557	264.4077	264.4697	264.5241
264.2614	263.6587	262.8828	261.8662	264.2903	264.3343
264.3913	264.4429	263.9871	263.3820	262.6789	261.5756
264.2148	264.2473	264.2959	264.3416	263.8403	263.1419
262.4546	261.2665	264.1395	264.1601	264.1958	264.2306
263.4653	262.7014	261.8358	260.6705	264.0732	264.0836
264.1050	263.6771	262.8172	262.1637	261.2090	260.0663
257.7952	257.8350	257.8810	257.9247	257.6298	257.1291
256.5143	255.7238	257.6640	257.6986	257.7379	257.7763
257.6376	257.0374	256.3115	255.4545	257.5709	257.6006
257.6351	257.6693	257.5257	257.0179	256.1452	255.2180
257.4946	257.5205	257.5535	257.5882	257.3195	256.7186
255.9489	254.9432	257.4181	257.4400	257.4720	257.5099
257.0507	256.4494	255.7616	254.6775	257.3361	257.3520
257.3817	257.4248	256.9299	256.2420	255.5757	254.4105
257.2528	257.2622	257.2916	257.3397	256.5916	255.8442
255.0034	253.8647	257.1755	257.1841	257.2165	256.8111
255.9711	255.3451	254.4265	253.3188	251.2751	251.3017
251.3239	251.3340	250.9987	250.4543	249.7914	248.9521
251.1309	251.1524	251.1705	251.1850	251.0145	250.3753
249.6033	248.6974	251.0259	251.0435	251.0589	251.0722
250.9022	250.3571	249.4395	248.4625	250.9450	250.9611
250.9772	250.9895	250.6911	250.0550	249.2403	248.1801

250.8711 250.8875 250.9057 250.9171 250.4233 249.7841

249.0463 247.8988 250.7964 250.8127 250.8318 250.8454

250.3061 249.5720 248.8463 247.6079 250.7196 250.7345

250.7546 250.7640 249.9633 249.1602 248.2547 247.0358

250.6417 250.6581 250.6735 250.2227 249.3280 248.6415

247.6534 246.4598 243.3839 243.4028 243.4112 243.3958

243.0274 242.4481 241.7503 240.8847 243.2598 243.2756

243.2831 243.2811 243.0772 242.3985 241.5849 240.6470

243.1619 243.1755 243.1821 243.1797 242.9805 242.3904

241.4285 240.4182 243.0844 243.0984 243.1068 243.0956

242.7636 242.0885 241.2258 240.1331 243.0168 243.0335

243.0455 243.0238 242.4903 241.8105 241.0220 239.8436

242.9518 242.9722 242.9873 242.9617 242.3699 241.5889

240.8107 239.5450 242.8861 242.9109 242.9247 242.8808

242.0245 241.1683 240.2166 238.9730 242.8178 242.8477

242.8401 242.3338 241.3862 240.6420 239.6076 238.3877

236.2257 236.2276 236.2162 236.1892 235.8369 235.2958

234.6581 233.8776 236.0837 236.0843 236.0755 236.0595

235.8652 235.2320 234.4870 233.6443 235.9688 235.9677

235.9596 235.9458 235.7554 235.2101 234.3270 233.4222

235.8804 235.8796 235.8730 235.8548 235.5405 234.9087

234.1224 233.1507 235.8106 235.8123 235.8081 235.7825

235.2763 234.6370 233.9260 232.8854 235.7501 235.7559

235.7541 235.7256 235.1673 234.4338 233.7392 232.6232

235.6918 235.7027 235.7010 235.6613 234.8517 234.0532

233.1979 232.1032 235.6301 235.6471 235.6340 235.1546

234.2556 233.5736 232.6499 231.5796 230.5345 230.5278

230.4980 230.4635 230.1431 229.6638 229.1061 228.4175

230.3647 230.3575 230.3315 230.3022 230.1344 229.5749

228.9234 228.1830 230.2175 230.2079 230.1833 230.1585

229.9929	229.5245	228.7499	227.9607	230.1024	230.0902
230.0647	230.0435	229.7668	229.2105	228.5343	227.6952
230.0145	230.0016	229.9752	229.9554	229.5015	228.9360
228.3369	227.4467	229.9435	229.9324	229.9060	229.8896
229.3973	228.7451	228.1669	227.2150	229.8774	229.8692
229.8489	229.8412	229.1133	228.4051	227.6735	226.7421
229.8085	229.8076	229.8005	229.3824	228.5657	227.9761
227.1881	226.2777				
0.0079	0.0078	0.0077	0.0074	0.0073	0.0065
0.0056	0.0050	0.0079	0.0078	0.0076	0.0074
0.0072	0.0064	0.0055	0.0049	0.0079	0.0078
0.0076	0.0073	0.0072	0.0065	0.0055	0.0048
0.0079	0.0077	0.0075	0.0073	0.0067	0.0059
0.0052	0.0047	0.0078	0.0077	0.0074	0.0071
0.0063	0.0055	0.0049	0.0046	0.0078	0.0076
0.0074	0.0070	0.0060	0.0052	0.0047	0.0045
0.0076	0.0074	0.0073	0.0068	0.0058	0.0051
0.0046	0.0044	0.0075	0.0073	0.0071	0.0063
0.0055	0.0049	0.0046	0.0044	0.0079	0.0078
0.0076	0.0074	0.0071	0.0063	0.0055	0.0050
0.0079	0.0078	0.0076	0.0073	0.0071	0.0063
0.0054	0.0049	0.0079	0.0077	0.0075	0.0073
0.0070	0.0063	0.0054	0.0048	0.0079	0.0077
0.0075	0.0072	0.0065	0.0057	0.0051	0.0047
0.0078	0.0076	0.0074	0.0070	0.0062	0.0053
0.0048	0.0046	0.0077	0.0075	0.0073	0.0069
0.0059	0.0051	0.0047	0.0045	0.0076	0.0074
0.0072	0.0067	0.0057	0.0050	0.0047	0.0044
0.0074	0.0073	0.0070	0.0062	0.0054	0.0049
0.0046	0.0044	0.0079	0.0078	0.0076	0.0074

0.0068	0.0060	0.0054	0.0051	0.0079	0.0077
0.0076	0.0073	0.0067	0.0059	0.0053	0.0050
0.0079	0.0077	0.0075	0.0072	0.0065	0.0058
0.0052	0.0049	0.0078	0.0077	0.0074	0.0069
0.0062	0.0055	0.0051	0.0048	0.0078	0.0076
0.0074	0.0067	0.0060	0.0054	0.0050	0.0047
0.0077	0.0075	0.0073	0.0066	0.0058	0.0052
0.0049	0.0046	0.0076	0.0074	0.0071	0.0064
0.0057	0.0052	0.0048	0.0046	0.0074	0.0072
0.0068	0.0062	0.0056	0.0051	0.0048	0.0045
0.0078	0.0078	0.0076	0.0073	0.0068	0.0061
0.0056	0.0053	0.0078	0.0077	0.0075	0.0073
0.0066	0.0060	0.0055	0.0051	0.0078	0.0077
0.0075	0.0071	0.0065	0.0059	0.0054	0.0050
0.0078	0.0077	0.0074	0.0069	0.0063	0.0057
0.0053	0.0049	0.0078	0.0076	0.0073	0.0068
0.0061	0.0056	0.0052	0.0049	0.0077	0.0075
0.0072	0.0066	0.0060	0.0055	0.0051	0.0048
0.0076	0.0074	0.0070	0.0065	0.0059	0.0054
0.0050	0.0047	0.0074	0.0072	0.0068	0.0062
0.0057	0.0053	0.0049	0.0046	0.0078	0.0077
0.0076	0.0073	0.0069	0.0064	0.0059	0.0056
0.0078	0.0077	0.0075	0.0072	0.0067	0.0062
0.0058	0.0054	0.0078	0.0077	0.0075	0.0071
0.0066	0.0061	0.0056	0.0053	0.0078	0.0076
0.0074	0.0070	0.0064	0.0059	0.0055	0.0052
0.0078	0.0076	0.0073	0.0068	0.0063	0.0059
0.0054	0.0051	0.0077	0.0075	0.0072	0.0067
0.0062	0.0058	0.0053	0.0050	0.0076	0.0073
0.0070	0.0066	0.0062	0.0057	0.0053	0.0049

0.0074 0.0072 0.0068 0.0064 0.0060 0.0056

0.0052 0.0047 0.0078 0.0077 0.0076 0.0073

0.0069 0.0065 0.0061 0.0057 0.0078 0.0077

0.0075 0.0072 0.0068 0.0064 0.0060 0.0056

0.0078 0.0077 0.0075 0.0071 0.0067 0.0063

0.0059 0.0055 0.0078 0.0076 0.0074 0.0070

0.0066 0.0062 0.0058 0.0054 0.0077 0.0076

0.0073 0.0069 0.0065 0.0061 0.0057 0.0053

0.0077 0.0075 0.0071 0.0068 0.0064 0.0060

0.0056 0.0051 0.0075 0.0073 0.0070 0.0066

0.0062 0.0058 0.0054 0.0050 0.0074 0.0071

0.0068 0.0064 0.0061 0.0057 0.0052 0.0047

0.0078 0.0077 0.0075 0.0072 0.0068 0.0063

0.0059 0.0054 0.0078 0.0077 0.0075 0.0072

0.0068 0.0063 0.0059 0.0054 0.0078 0.0077

0.0074 0.0071 0.0067 0.0063 0.0058 0.0054

0.0078 0.0076 0.0074 0.0070 0.0066 0.0062

0.0058 0.0053 0.0077 0.0075 0.0072 0.0069

0.0065 0.0062 0.0058 0.0053 0.0076 0.0074

0.0071 0.0068 0.0064 0.0061 0.0057 0.0052

0.0075 0.0073 0.0070 0.0066 0.0063 0.0059

0.0056 0.0050 0.0073 0.0071 0.0068 0.0064

0.0061 0.0058 0.0054 0.0048 0.0077 0.0077

0.0074 0.0069 0.0062 0.0056 0.0050 0.0045

0.0077 0.0076 0.0074 0.0069 0.0062 0.0056

0.0050 0.0045 0.0077 0.0076 0.0073 0.0068

0.0063 0.0057 0.0051 0.0045 0.0077 0.0076

0.0072 0.0067 0.0062 0.0056 0.0050 0.0044

0.0076 0.0075 0.0071 0.0066 0.0060 0.0055

0.0049 0.0042 0.0076 0.0074 0.0069 0.0064

0.0059 0.0053 0.0047 0.0041 0.0075 0.0072

0.0067 0.0062 0.0057 0.0051 0.0045 0.0039

0.0073 0.0069 0.0065 0.0060 0.0055 0.0050

0.0044 0.0037 0.0066 0.0065 0.0062 0.0056

0.0050 0.0044 0.0039 0.0034 0.0065 0.0064

0.0061 0.0056 0.0051 0.0045 0.0039 0.0035

0.0064 0.0063 0.0060 0.0056 0.0051 0.0046

0.0040 0.0036 0.0063 0.0061 0.0058 0.0055

0.0051 0.0046 0.0041 0.0036 0.0061 0.0060

0.0057 0.0054 0.0051 0.0046 0.0041 0.0036

0.0059 0.0058 0.0056 0.0054 0.0051 0.0046

0.0041 0.0035 0.0057 0.0056 0.0055 0.0053

0.0049 0.0044 0.0039 0.0033 0.0055 0.0055

0.0054 0.0051 0.0047 0.0042 0.0037 0.0031

0.0027 0.0027 0.0026 0.0026 0.0026 0.0025

0.0024 0.0022 0.0025 0.0024 0.0024 0.0025

0.0025 0.0025 0.0024 0.0022 0.0023 0.0023

0.0024 0.0025 0.0026 0.0026 0.0025 0.0023

0.0023 0.0024 0.0025 0.0027 0.0028 0.0028

0.0026 0.0024 0.0025 0.0026 0.0028 0.0030

0.0030 0.0029 0.0027 0.0025 0.0028 0.0029

0.0031 0.0033 0.0032 0.0031 0.0029 0.0026

0.0031 0.0032 0.0034 0.0035 0.0034 0.0031

0.0029 0.0026 0.0034 0.0036 0.0037 0.0036

0.0034 0.0031 0.0028 0.0026 0.0019 0.0020

0.0020 0.0020 0.0020 0.0019 0.0019 0.0019

0.0020 0.0020 0.0021 0.0021 0.0021 0.0021

0.0020 0.0020 0.0021 0.0021 0.0022 0.0022

0.0022 0.0022 0.0021 0.0020 0.0022 0.0022

0.0023 0.0023 0.0023 0.0022 0.0021 0.0020

0.0023	0.0023	0.0023	0.0024	0.0023	0.0022
0.0022	0.0021	0.0024	0.0024	0.0025	0.0025
0.0024	0.0023	0.0022	0.0021	0.0025	0.0025
0.0026	0.0026	0.0025	0.0024	0.0022	0.0021
0.0026	0.0027	0.0027	0.0026	0.0025	0.0024
0.0023	0.0022	0.0016	0.0015	0.0015	0.0015
0.0015	0.0016	0.0016	0.0016	0.0016	0.0016
0.0015	0.0015	0.0015	0.0016	0.0016	0.0017
0.0017	0.0016	0.0016	0.0016	0.0016	0.0016
0.0017	0.0017	0.0017	0.0017	0.0016	0.0016
0.0016	0.0017	0.0017	0.0017	0.0018	0.0018
0.0017	0.0017	0.0017	0.0017	0.0017	0.0017
0.0019	0.0018	0.0018	0.0018	0.0018	0.0018
0.0018	0.0018	0.0020	0.0020	0.0019	0.0019
0.0019	0.0019	0.0018	0.0018	0.0021	0.0021
0.0020	0.0020	0.0020	0.0019	0.0019	0.0019
0.0018	0.0018	0.0018	0.0018	0.0018	0.0019
0.0018	0.0018	0.0018	0.0018	0.0019	0.0019
0.0019	0.0019	0.0019	0.0019	0.0019	0.0019
0.0019	0.0019	0.0019	0.0019	0.0019	0.0019
0.0019	0.0019	0.0020	0.0020	0.0020	0.0020
0.0020	0.0019	0.0020	0.0020	0.0020	0.0020
0.0020	0.0020	0.0020	0.0019	0.0020	0.0021
0.0021	0.0021	0.0021	0.0020	0.0020	0.0019
0.0021	0.0021	0.0021	0.0021	0.0021	0.0020
0.0020	0.0019	0.0021	0.0021	0.0021	0.0021
0.0021	0.0020	0.0019	0.0018	0.0016	0.0016
0.0016	0.0015	0.0015	0.0015	0.0014	0.0013
0.0016	0.0016	0.0016	0.0015	0.0015	0.0015
0.0014	0.0013	0.0016	0.0016	0.0016	0.0015

0.0015	0.0014	0.0014	0.0013	0.0016	0.0016
0.0016	0.0015	0.0015	0.0014	0.0014	0.0013
0.0016	0.0016	0.0016	0.0015	0.0015	0.0014
0.0014	0.0013	0.0016	0.0016	0.0016	0.0016
0.0015	0.0014	0.0014	0.0013	0.0016	0.0016
0.0016	0.0016	0.0015	0.0014	0.0014	0.0012
0.0016	0.0016	0.0016	0.0016	0.0015	0.0014
0.0013	0.0012	0.0012	0.0012	0.0012	0.0012
0.0011	0.0011	0.0010	0.0010	0.0012	0.0012
0.0012	0.0012	0.0012	0.0011	0.0010	0.0010
0.0012	0.0012	0.0012	0.0012	0.0012	0.0011
0.0011	0.0010	0.0012	0.0012	0.0012	0.0012
0.0012	0.0011	0.0011	0.0010	0.0013	0.0012
0.0012	0.0012	0.0012	0.0011	0.0011	0.0010
0.0013	0.0012	0.0012	0.0012	0.0012	0.0011
0.0011	0.0010	0.0013	0.0013	0.0012	0.0012
0.0012	0.0011	0.0010	0.0010	0.0013	0.0013
0.0012	0.0012	0.0011	0.0011	0.0010	0.0010
0.0008	0.0008	0.0008	0.0008	0.0008	0.0007
0.0007	0.0007	0.0009	0.0008	0.0008	0.0008
0.0008	0.0008	0.0007	0.0007	0.0009	0.0009
0.0009	0.0008	0.0008	0.0008	0.0007	0.0007
0.0009	0.0009	0.0009	0.0008	0.0008	0.0008
0.0007	0.0007	0.0009	0.0009	0.0009	0.0009
0.0008	0.0008	0.0007	0.0007	0.0009	0.0009
0.0009	0.0009	0.0008	0.0008	0.0007	0.0007
0.0009	0.0009	0.0009	0.0009	0.0008	0.0008
0.0007	0.0007	0.0009	0.0009	0.0009	0.0009
0.0008	0.0008	0.0007	0.0006	0.0005	0.0005
0.0005	0.0005	0.0005	0.0005	0.0004	0.0004

0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
0.0004	0.0004	0.0006	0.0005	0.0005	0.0005
0.0005	0.0005	0.0004	0.0004	0.0006	0.0006
0.0006	0.0005	0.0005	0.0005	0.0005	0.0004
0.0006	0.0006	0.0006	0.0006	0.0005	0.0005
0.0005	0.0004	0.0006	0.0006	0.0006	0.0006
0.0005	0.0005	0.0005	0.0004	0.0006	0.0006
0.0006	0.0006	0.0005	0.0005	0.0005	0.0004
0.0006	0.0006	0.0006	0.0006	0.0005	0.0005
0.0004	0.0004	0.0003	0.0003	0.0003	0.0003
0.0002	0.0002	0.0002	0.0002	0.0003	0.0003
0.0003	0.0003	0.0003	0.0002	0.0002	0.0002
0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
0.0002	0.0002	0.0003	0.0003	0.0003	0.0003
0.0003	0.0003	0.0002	0.0002	0.0003	0.0003
0.0003	0.0003	0.0003	0.0003	0.0002	0.0002
0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
0.0002	0.0002	0.0003	0.0003	0.0003	0.0003
0.0003	0.0003	0.0002	0.0002	0.0003	0.0003
0.0003	0.0003	0.0003	0.0003	0.0002	0.0002
0.0002	0.0002	0.0002	0.0002	0.0002	0.0001
0.0001	0.0001	0.0002	0.0002	0.0002	0.0002
0.0002	0.0001	0.0001	0.0001	0.0002	0.0002
0.0002	0.0002	0.0002	0.0002	0.0001	0.0001
0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
0.0001	0.0001	0.0002	0.0002	0.0002	0.0002
0.0002	0.0002	0.0001	0.0001	0.0002	0.0002
0.0002	0.0002	0.0002	0.0002	0.0001	0.0001
0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
0.0001	0.0001	0.0002	0.0002	0.0002	0.0002

0.0002 0.0001 0.0001 0.0001 0.0000 0.0000

0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

0.0001 0.0000 0.0000 0.0000 0.0000 0.0000

0.0000 0.0000 0.0001 0.0000 0.0000 0.0000

0.0000 0.0000 0.0000 0.0001 0.0001 0.0001

0.0000 0.0000 0.0000 0.0000 0.0001 0.0001

0.0001 0.0000 0.0000 0.0000 0.0000 0.0000

0.0001 0.0001 0.0001 0.0000 0.0000 0.0000

0.0000 0.0000

10.0061 10.0061 10.0061 10.0061 9.9897 9.9636

9.9322 9.8923 10.0061 10.0061 10.0061 10.0061

9.9975 9.9661 9.9289 9.8854 10.0061 10.0061

10.0061 10.0061 9.9972 9.9703 9.9256 9.8785

10.0061 10.0061 10.0061 10.0061 9.9908 9.9592

9.9197 9.8686 10.0061 10.0061 10.0061 10.0061

9.9810 9.9494 9.9138 9.8586 10.0061 10.0061

10.0061 10.0061 9.9788 9.9426 9.9079 9.8487

10.0061 10.0061 10.0061 10.0061 9.9658 9.9267

9.8832 9.8255 10.0061 10.0061 10.0061 9.9831

9.9387 9.9059 9.8585 9.8023 30.0184 30.0184

30.0184 30.0184 29.9691 29.8907 29.7965 29.6768

30.0184 30.0184 30.0184 30.0184 29.9926 29.8982

29.7866 29.6562 30.0184 30.0184 30.0184 30.0184

29.9915 29.9110 29.7767 29.6356 30.0184 30.0184

30.0184 30.0184 29.9724 29.8776 29.7590 29.6057

30.0184 30.0184 30.0184 30.0184 29.9430 29.8482

29.7414 29.5759 30.0184 30.0184 30.0184 30.0184

29.9365	29.8279	29.7237	29.5461	30.0184	30.0184
30.0184	30.0184	29.8973	29.7800	29.6496	29.4765
30.0184	30.0184	30.0184	29.9492	29.8162	29.7177
29.5756	29.4069	55.0337	55.0337	55.0337	55.0337
54.9433	54.7995	54.6268	54.4075	55.0337	55.0337
55.0337	55.0337	54.9865	54.8134	54.6087	54.3697
55.0337	55.0337	55.0337	55.0337	54.9844	54.8368
54.5906	54.3318	55.0337	55.0337	55.0337	55.0337
54.9494	54.7756	54.5582	54.2772	55.0337	55.0337
55.0337	55.0337	54.8955	54.7217	54.5258	54.2225
55.0337	55.0337	55.0337	55.0337	54.8836	54.6844
54.4934	54.1678	55.0337	55.0337	55.0337	55.0337
54.8118	54.5967	54.3576	54.0402	55.0337	55.0337
55.0337	54.9068	54.6631	54.4825	54.2219	53.9126
90.0551	90.0551	90.0551	90.0551	89.9072	89.6720
89.3894	89.0305	90.0551	90.0551	90.0551	90.0551
89.9778	89.6946	89.3597	88.9686	90.0551	90.0551
90.0551	90.0551	89.9745	89.7329	89.3301	88.9067
90.0551	90.0551	90.0551	90.0551	89.9172	89.6328
89.2771	88.8172	90.0551	90.0551	90.0551	90.0551
89.8290	89.5447	89.2241	88.7277	90.0551	90.0551
90.0551	90.0551	89.8096	89.4836	89.1710	88.6382
90.0551	90.0551	90.0551	90.0551	89.6921	89.3400
88.9489	88.4294	90.0551	90.0551	90.0551	89.8475
89.4487	89.1532	88.7268	88.2207	140.0857	140.0857
140.0857	140.0857	139.8556	139.4897	139.0502	138.4918
140.0857	140.0857	140.0857	140.0857	139.9655	139.5250
139.0041	138.3956	140.0857	140.0857	140.0857	140.0857
139.9604	139.5845	138.9580	138.2992	140.0857	140.0857
140.0857	140.0857	139.8711	139.4287	138.8755	138.1600

140.0857	140.0857	140.0857	140.0857	139.7340	139.2917
138.7930	138.0208	140.0857	140.0857	140.0857	140.0857
139.7038	139.1967	138.7105	137.8816	140.0857	140.0857
140.0857	140.0857	139.5210	138.9734	138.3649	137.5569
140.0857	140.0857	140.0857	139.7627	139.1424	138.6828
138.0194	137.2322	215.1316	215.1316	215.1316	215.1316
214.7782	214.2164	213.5413	212.6839	215.1316	215.1316
215.1316	215.1316	214.9471	214.2705	213.4705	212.5360
215.1316	215.1316	215.1316	215.1316	214.9392	214.3619
213.3997	212.3881	215.1316	215.1316	215.1316	215.1316
214.8021	214.1227	213.2731	212.1743	215.1316	215.1316
215.1316	215.1316	214.5915	213.9123	213.1463	211.9606
215.1316	215.1316	215.1316	215.1316	214.5451	213.7663
213.0196	211.7468	215.1316	215.1316	215.1316	215.1316
214.2643	213.4234	212.4890	211.2481	215.1316	215.1316
215.1316	214.6356	213.6830	212.9772	211.9584	210.7494
330.2020	330.2020	330.2020	330.2020	329.6596	328.7973
327.7610	326.4451	330.2020	330.2020	330.2020	330.2020
329.9187	328.8804	327.6524	326.2181	330.2020	330.2020
330.2020	330.2020	329.9066	329.0206	327.5438	325.9911
330.2020	330.2020	330.2020	330.2020	329.6962	328.6534
327.3493	325.6630	330.2020	330.2020	330.2020	330.2020
329.3730	328.3305	327.1549	325.3348	330.2020	330.2020
330.2020	330.2020	329.3018	328.1065	326.9604	325.0067
330.2020	330.2020	330.2020	330.2020	328.8708	327.5801
326.1459	324.2413	330.2020	330.2020	330.2020	329.4407
327.9785	326.8952	325.3314	323.4759	500.3061	500.3061
500.3061	500.3061	499.4842	498.1776	496.6077	494.6137
500.3061	500.3061	500.3061	500.3061	499.8769	498.3036
496.4431	494.2698	500.3061	500.3061	500.3061	500.3061

499.8586	498.5160	496.2785	493.9259	500.3061	500.3061
500.3061	500.3061	499.5398	497.9597	495.9838	493.4287
500.3061	500.3061	500.3061	500.3061	499.0499	497.4704
495.6892	492.9316	500.3061	500.3061	500.3061	500.3061
498.9421	497.1310	495.3944	492.4344	500.3061	500.3061
500.3061	500.3061	498.2892	496.3335	494.1604	491.2747
500.3061	500.3061	500.3061	499.1526	496.9371	495.2958
492.9264	490.1149	750.4592	750.4592	750.4592	750.4592
749.2264	747.2665	744.9115	741.9207	750.4592	750.4592
750.4592	750.4592	749.8153	747.4554	744.6646	741.4048
750.4592	750.4592	750.4592	750.4592	749.7879	747.7740
744.4177	740.8889	750.4592	750.4592	750.4592	750.4592
749.3097	746.9396	743.9758	740.1431	750.4592	750.4592
750.4592	750.4592	748.5749	746.2056	743.5338	739.3974
750.4592	750.4592	750.4592	750.4592	748.4131	745.6965
743.0917	738.6517	750.4592	750.4592	750.4592	750.4592
747.4337	744.5002	741.2406	736.9120	750.4592	750.4592
750.4592	748.7289	745.4056	742.9437	739.3896	735.1724
1100.6735	1100.6735	1100.6735	1100.6735	1098.8652	1095.9908
1092.5369	1088.1503	1100.6735	1100.6735	1100.6735	1100.6735
1099.7291	1096.2681	1092.1748	1087.3937	1100.6735	1100.6735
1100.6735	1100.6735	1099.6890	1096.7352	1091.8127	1086.6370
1100.6735	1100.6735	1100.6735	1100.6735	1098.9874	1095.5115
1091.1646	1085.5432	1100.6735	1100.6735	1100.6735	1100.6735
1097.9099	1094.4348	1090.5161	1084.4495	1100.6735	1100.6735
1100.6735	1100.6735	1097.6726	1093.6882	1089.8678	1083.3558
1100.6735	1100.6735	1100.6735	1100.6735	1096.2361	1091.9337
1087.1530	1080.8042	1100.6735	1100.6735	1100.6735	1098.1357
1093.2617	1089.6508	1084.4381	1078.2528	1600.9795	1600.9795
1600.9795	1600.9795	1598.3495	1594.1686	1589.1445	1582.7640

1600.9795 1600.9795 1600.9795 1600.9795 1599.6060 1594.5714
1588.6178 1581.6636 1600.9795 1600.9795 1600.9795 1600.9795
1599.5474 1595.2513 1588.0913 1580.5627 1600.9795 1600.9795
1600.9795 1600.9795 1598.5272 1593.4712 1587.1483 1578.9720
1600.9795 1600.9795 1600.9795 1600.9795 1596.9598 1591.9054
1586.2054 1577.3811 1600.9795 1600.9795 1600.9795 1600.9795
1596.6147 1590.8191 1585.2622 1575.7903 1600.9795 1600.9795
1600.9795 1600.9795 1594.5254 1588.2672 1581.3134 1572.0790
1600.9795 1600.9795 1600.9795 1597.2883 1590.1987 1584.9465
1577.3645 1568.3679 2301.4080 2301.4080 2301.4080 2301.4080
2297.6274 2291.6174 2284.3955 2275.2234 2301.4080 2301.4080
2301.4080 2301.4080 2299.4338 2292.1965 2283.6382 2273.6411
2301.4080 2301.4080 2301.4080 2301.4080 2299.3494 2293.1738
2282.8811 2272.0591 2301.4080 2301.4080 2301.4080 2301.4080
2297.8831 2290.6147 2281.5259 2269.7722 2301.4080 2301.4080
2301.4080 2301.4080 2295.6299 2288.3640 2280.1702 2267.4851
2301.4080 2301.4080 2301.4080 2301.4080 2295.1338 2286.8025
2278.8142 2265.1987 2301.4080 2301.4080 2301.4080 2301.4080
2292.1304 2283.1340 2273.1379 2259.8635 2301.4080 2301.4080
2301.4080 2296.1021 2285.9106 2278.3606 2267.4614 2254.5288
3101.8979 3101.8979 3101.8979 3101.8979 3096.8022 3088.7017
3078.9675 3066.6052 3101.8979 3101.8979 3101.8979 3101.8979
3099.2368 3089.4824 3077.9473 3064.4729 3101.8979 3101.8979
3101.8979 3101.8979 3099.1233 3090.7996 3076.9268 3062.3401
3101.8979 3101.8979 3101.8979 3101.8979 3097.1465 3087.3503
3075.1001 3059.2581 3101.8979 3101.8979 3101.8979 3101.8979
3094.1096 3084.3167 3073.2727 3056.1758 3101.8979 3101.8979
3101.8979 3101.8979 3093.4412 3082.2122 3071.4456 3053.0938
3101.8979 3101.8979 3101.8979 3101.8979 3089.3926 3077.2676
3063.7947 3045.9031 3101.8979 3101.8979 3101.8979 3094.7461

3081.0098 3070.8337 3056.1436 3038.7124 3902.3875 3902.3875
3902.3875 3902.3875 3895.9766 3885.7859 3873.5396 3857.9873
3902.3875 3902.3875 3902.3875 3902.3875 3899.0398 3886.7681
3872.2561 3855.3049 3902.3875 3902.3875 3902.3875 3902.3875
3898.8967 3888.4250 3870.9724 3852.6218 3902.3875 3902.3875
3902.3875 3902.3875 3896.4102 3884.0859 3868.6741 3848.7444
3902.3875 3902.3875 3902.3875 3902.3875 3892.5891 3880.2693
3866.3755 3844.8660 3902.3875 3902.3875 3902.3875 3902.3875
3891.7485 3877.6218 3864.0767 3840.9888 3902.3875 3902.3875
3902.3875 3902.3875 3886.6558 3871.4014 3854.4514 3831.9424
3902.3875 3902.3875 3902.3875 3893.3904 3876.1096 3863.3071
3844.8262 3822.8962 4802.9385 4802.9385 4802.9385 4802.9385
4795.0483 4782.5054 4767.4331 4748.2920 4802.9385 4802.9385
4802.9385 4802.9385 4798.8179 4783.7144 4765.8535 4744.9902
4802.9385 4802.9385 4802.9385 4802.9385 4798.6421 4785.7534
4764.2734 4741.6885 4802.9385 4802.9385 4802.9385 4802.9385
4795.5820 4780.4136 4761.4448 4736.9165 4802.9385 4802.9385
4802.9385 4802.9385 4790.8799 4775.7158 4758.6162 4732.1431
4802.9385 4802.9385 4802.9385 4802.9385 4789.8442 4772.4580
4755.7871 4727.3706 4802.9385 4802.9385 4802.9385 4802.9385
4783.5762 4764.8013 4743.9399 4716.2368 4802.9385 4802.9385
4802.9385 4791.8647 4770.5962 4754.8398 4732.0933 4705.1030
5803.5508 5803.5508 5803.5508 5803.5508 5794.0171 5778.8613
5760.6484 5737.5200 5803.5508 5803.5508 5803.5508 5803.5508
5798.5718 5780.3218 5758.7397 5733.5298 5803.5508 5803.5508
5803.5508 5803.5508 5798.3599 5782.7861 5756.8306 5729.5405
5803.5508 5803.5508 5803.5508 5803.5508 5794.6611 5776.3330
5753.4126 5723.7734 5803.5508 5803.5508 5803.5508 5803.5508
5788.9795 5770.6572 5749.9946 5718.0063 5803.5508 5803.5508
5803.5508 5803.5508 5787.7285 5766.7197 5746.5757 5712.2397

5803.5508	5803.5508	5803.5508	5803.5508	5780.1548	5757.4683
5732.2612	5698.7861	5803.5508	5803.5508	5803.5508	5790.1699
5764.4702	5745.4312	5717.9463	5685.3330	6804.1626	6804.1626
6804.1626	6804.1626	6792.9854	6775.2168	6753.8638	6726.7476
6804.1626	6804.1626	6804.1626	6804.1626	6798.3257	6776.9292
6751.6255	6722.0693	6804.1626	6804.1626	6804.1626	6804.1626
6798.0771	6779.8179	6749.3877	6717.3916	6804.1626	6804.1626
6804.1626	6804.1626	6793.7412	6772.2520	6745.3804	6710.6304
6804.1626	6804.1626	6804.1626	6804.1626	6787.0791	6765.5977
6741.3726	6703.8696	6804.1626	6804.1626	6804.1626	6804.1626
6785.6128	6760.9819	6737.3652	6697.1084	6804.1626	6804.1626
6804.1626	6804.1626	6776.7329	6750.1353	6720.5820	6681.3359
6804.1626	6804.1626	6804.1626	6788.4746	6758.3442	6736.0225
6703.7983	6665.5630	7804.7749	7804.7749	7804.7749	7804.7749
7791.9531	7771.5718	7747.0791	7715.9746	7804.7749	7804.7749
7804.7749	7804.7749	7798.0796	7773.5361	7744.5122	7710.6099
7804.7749	7804.7749	7804.7749	7804.7749	7797.7935	7776.8501
7741.9448	7705.2437	7804.7749	7804.7749	7804.7749	7804.7749
7792.8203	7768.1719	7737.3481	7697.4888	7804.7749	7804.7749
7804.7749	7804.7749	7785.1782	7760.5386	7732.7510	7689.7319
7804.7749	7804.7749	7804.7749	7804.7749	7783.4971	7755.2437
7728.1533	7681.9775	7804.7749	7804.7749	7804.7749	7804.7749
7773.3115	7742.8027	7708.9028	7663.8848	7804.7749	7804.7749
7804.7749	7786.7808	7752.2192	7726.6143	7689.6523	7645.7925
8680.3105	8680.3105	8680.3105	8680.3105	8666.0518	8643.3828
8616.1436	8581.5479	8680.3105	8680.3105	8680.3105	8680.3105
8672.8643	8645.5664	8613.2871	8575.5820	8680.3105	8680.3105
8680.3105	8680.3105	8672.5459	8649.2529	8610.4316	8569.6133
8680.3105	8680.3105	8680.3105	8680.3105	8667.0146	8639.6016
8605.3193	8560.9893	8680.3105	8680.3105	8680.3105	8680.3105

8658.5166 8631.1123 8600.2070 8552.3623 8680.3105 8680.3105
8680.3105 8680.3105 8656.6455 8625.2227 8595.0938 8543.7383
8680.3105 8680.3105 8680.3105 8680.3105 8645.3174 8611.3867
8573.6826 8523.6172 8680.3105 8680.3105 8680.3105 8660.2979
8621.8584 8593.3818 8552.2734 8503.4941 9430.7695 9430.7695
9430.7695 9430.7695 9415.2783 9390.6494 9361.0547 9323.4688
9430.7695 9430.7695 9430.7695 9430.7695 9422.6787 9393.0234
9357.9521 9316.9854 9430.7695 9430.7695 9430.7695 9430.7695
9422.3340 9397.0283 9354.8496 9310.5029 9430.7695 9430.7695
9430.7695 9430.7695 9416.3252 9386.5410 9349.2949 9301.1318
9430.7695 9430.7695 9430.7695 9430.7695 9407.0918 9377.3184
9343.7412 9291.7598 9430.7695 9430.7695 9430.7695 9430.7695
9405.0586 9370.9199 9338.1865 9282.3887 9430.7695 9430.7695
9430.7695 9430.7695 9392.7510 9355.8867 9314.9238 9260.5264
9430.7695 9430.7695 9430.7695 9409.0264 9367.2646 9336.3252
9291.6631 9238.6670

0.0000 0.0000 0.0000 0.0000 57.1678 148.0451
257.2551 395.9436 0.0000 0.0000 0.0000 0.0000
29.8558 139.2903 268.7019 419.8674 0.0000 0.0000
0.0000 0.0000 31.1276 124.5106 280.1488 443.7913
0.0000 0.0000 0.0000 0.0000 53.3033 163.2054
300.6454 478.3713 0.0000 0.0000 0.0000 0.0000
87.3732 197.2409 321.1422 512.9514 0.0000 0.0000
0.0000 0.0000 94.8756 220.8526 341.6389 547.5314
0.0000 0.0000 0.0000 0.0000 140.2942 276.3247
427.4756 628.2017 0.0000 0.0000 0.0000 80.2348
234.3408 348.5021 513.3124 708.8719

APPENDIX B: TEXT OF PROGRAM

A. JAVAWEATHER.JAVA

```
import java.io.*;

import java.net.*;

import java.text.*;

import java.util.*;

import java.util.regex.*;

import mil.navy.nrlssc.dmap.TEDServices.Constants.*;

import mil.navy.nrlssc.dmap.TEDServices.Enumerations.*;

import mil.navy.nrlssc.dmap.TEDServices.DatabaseInterfaceModule.*;

import mil.navy.nrlssc.dmap.TEDServices.DatabaseInterfaceModule.Common.*;

import mil.navy.nrlssc.dmap.TEDServices.DatabaseInterfaceModule.InterfacePkg.*;

import mil.navy.nrlssc.dmap.TEDServices.InterfaceSupport.*;

import mil.navy.nrlssc.dmap.TEDServices.TEDSServer.*;

import mil.navy.nrlssc.dmap.TEDServices.TedsTransmittalModel.*;


/**

 * JEMWeather.java

 *

 * <br>This interface sets all the base standards for JEMWeather

 *

 * @author LCDR Victor Ross, USN

 * @version 1.0

 *

 */


interface JEMConstant {


  /**
```

```
 * Default TEDServices.
 * Must be an IP address or a DNS entry
 */
public static String database = "207.85.236.26";

/**
 * Port Contact Number.
 * <br>Use 80 for unsecure<br>Use 443 for SSL
 */
public int port = 80;

/**
 * TEDServices User Name.
 */
public String uName = "abroc2000";

/**
 * TEDServices Password.
 */
public String uPass = "map.zxcv";

/**
 * TEDServices Height Coordinates.
 *   <br>Choices are MILLIBAR, COAMPS_SIGMA, DISTANCE
 */
public String htCoord = "METER";

/**
 * 3D Parameters to Retrieve - MUST BE IN ORDER SHOWN.
 *   Minimum Parameters for JEM/HPAC MEDOC format =
```

```
   *   String[]{
   *   <br>"U_WIND",
   *   <br>"V_WIND",
   *   <br>"W_WIND",
   *   <br>"POTENTIAL_TEMPERATURE",
   *   <br>"WATER_VAPOR_MIXING_RATIO",
   *   <br>"TOTAL_PRESSURE"};
   */
  public String[] strAttributeCodes3 = new String[]{"U_WIND", "V_WIND",
"W_WIND", "POTENTIAL_TEMPERATURE", "WATER_VAPOR_MIXING_RATIO",
"TOTAL_PRESSURE"};


  /**
   * 2D Parameters to Retrieve.
   *   <br>Minimum Parameters for JEM/HPAC MEDOC format =
   *   String[]{"TERRAIN_HEIGHT"};
   */
  public String[] strAttributeCodes2 = new String[]{"TERRAIN_HEIGHT"};


  /**
   * Tau increment in hours.
   * <br>example int tauInc = 1;
   */
  public int tauInc = 1;


  /**
   * Output Directory to Write MEDOC Formatted File.
   * <br>example String outputDir = "C:\Temp\";
   */
  public String outputDir = "/tmp/";
```

```
}



/**
 * JEMWeather.java
 *
 * <br>This class should pull basic data from TEDServices
 * for use by HPAC/JEM.
 *
 * @author LCDR Victor Ross, USN
 * @version 0.7
 */

public class JEMWeather implements JEMConstant{

  /**
   * Taus to Retrieve.
   *    This is collected from the database in real time.
   */
  public int[] taus;

  /**
   * Bounding Box.
   *   use float[]{North (Lat) ,South (Lat) ,East (Lon) ,West (Lon)};
   *   <br>Example float[]{35.0f,29.0f,-115.0f,-121.0f};
   */
  public float[] bBox = new float[4];

  /**
   * Hash Map of JEM/HPAC ID to TEDServices ID.
```

```
 */
public HashMap convertTable = new HashMap();

/**
 * PrintWriter to hold all output for printing to file.
 */
public PrintWriter pw;

/**
 * Holds the pressure data for calculation.
 */
public float[][][] presData;

/**
 * Default Constructor
 * @param inBox Float Array of Positions [N,S,E,W]
 * @param incidentTime String of the time of the WMD/WME
 */
public JEMWeather(float[] inBox, String incidentTime) {
  taus = tauCalc(incidentTime);
  // Set the database
  String host = database;
  // check that the lat lon values are good
  if (inBox[0] > 90 || inBox[0] <-90 || inBox[1] > 90 || inBox[1] <-90) {
    System.out.println("That Latitude does not exist!");
    System.exit(1);
  }
  if (inBox[2] > 180 || inBox[2] <-180 || inBox[3] > 180 || inBox[3] <-180) {
    System.out.println("That Longitude does not exist!");
    System.exit(1);
```

```
}
bBox = inBox;
// COMBINE 2D and 3D Attributes
/**
 * Combined Parameters to Retrieve.
 */
int totalAttributes = strAttributeCodes3.length+strAttributeCodes2.length;
String[] strAttributeCodes = new String[totalAttributes];
int tempi = 0;
for (int i=0; i<strAttributeCodes3.length; i++) {
  strAttributeCodes[tempi]=strAttributeCodes3[i];
  tempi++;
}
for (int i=0; i<strAttributeCodes2.length; i++) {
  strAttributeCodes[tempi]=strAttributeCodes2[i];
  tempi++;
}
// CREATE CROSSTABLE FOR JEM/HPAC and TEDServices
convertTable.put("U_WIND","U:M/S");
convertTable.put("V_WIND","V:M/S");
convertTable.put("W_WIND","W:M/S");
convertTable.put("POTENTIAL_TEMPERATURE","T:KELVIN");
convertTable.put("WATER_VAPOR_MIXING_RATIO","H:GM/GM");
//PULLED DATA AS A PLACE HOLDER
convertTable.put("TOTAL_PRESSURE","PHI:METERS");
convertTable.put("TERRAIN_HEIGHT","TOPO:METERS");
convertTable.put("PLANETARY_BOUNDARY_LAYER_HEIGHT","ZI:METERS");
convertTable.put("LATENT_HEAT_FLUX","HFLX:W/M2");
try {
  APIInterface api = new APIInterface(host, port, uName, uPass);
```

```
    GridParameters3D gridParameters = new GridParameters3D((byte)0,
                "ATMOSPHERIC_FORECAST",
                taus,
                strAttributeCodes,
                bBox[3],
                bBox[1],
                bBox[2],
                bBox[0],
                false  // set to false since not a subscription
                );
    Grid3D[] grids = api.getGrids(gridParameters);
    //showGrids(grids);
    writeMEDOC(grids);
  }
  catch(Exception e){
    System.err.println("Error in JEMWeather constructor: " + e);
    e.printStackTrace();
  }
} //finishes constructor


/**
 * Show Collected Grid Information.
 * @param gridsOut The grids collected from TEDService
 * Shows the information about the grids returned from TEDServices
 */
public void showGrids(Grid3D[] gridsOut) {
  try{
    System.out.println("retrieved " + gridsOut.length + " grid3d objects");
    for (int i = 0; i < gridsOut.length; i++){
      System.out.println("---------->got a 3d grid back<----------");
```

```
        System.out.println("grid parameter is " + gridsOut[i].parameterName);
        // extract grid information
        double xResolution = gridsOut[i].xAxisSpacing;
        double yResolution = gridsOut[i].yAxisSpacing;
        double[] zLevel = gridsOut[i].zAxisValueArray;
        bBox = gridsOut[i].getBoundingBox();
        // extract actual grid point values
        float[][][] gridData = gridsOut[i].data;
        int gL=gridsOut[i].data.length;
        int gW=gridsOut[i].data[0].length;
        int gH=gridsOut[i].data[0][0].length;
        System.out.println("grid has " + gL + " levels " + gW + " rows and " +
gH + " columns ");
        System.out.println("grid spacings are");
        System.out.println(" X:" + xResolution + " " +
gridsOut[i].horizontalAxesUnitName );
        System.out.println(" Y:" +yResolution + " "  +
gridsOut[i].horizontalAxesUnitName);
        System.out.print(" Z:");
        for (int j=0;j<zLevel.length; j++) {System.out.print("l"+j+"
"+zLevel[j]+"-");}
       System.out.println(" "+gridsOut[i].verticalAxesUnitName);
        System.out.println("output time period is " +
gridsOut[i].forecastRunTimeString);
        System.out.println("TAU is " + gridsOut[i].forecastEventTime);
        System.out.println("The origin is " + gridsOut[i].longitude + " " +
gridsOut[i].latitude);
        System.out.println("The borders are N:"+bBox[3]+" S:"+ bBox[1]+"
E:"+bBox[2]+" W:"+bBox[0]);

System.out.println("==========================================================
======");
```

```
      }
    } catch (Exception e) {
      System.out.println(e);
    }
  }

  /**
   * Write Collected Grid Information to File in MEDOC Format.
   * @param gridsOut The grids collected from TEDServices
   * Writes out the information about the grids returned from TEDServices
   */
  public void writeMEDOC(Grid3D[] gridsOut) {
   try {                          // Delete the old file if it exists
     File nFile = new File(outputDir+"JEM.fmt");
     if (nFile.exists()) {
       nFile.delete();
     }
     OutputStream fos = new FileOutputStream(outputDir+"JEM.fmt");
     pw = new PrintWriter(fos, true);
     DecimalFormat form = new DecimalFormat("#0.0000"); // Set up the Number
Formatting
     int numberGrids = gridsOut.length;                // run the loops to
write out the required MEDOC DATA
     for (int t=0; t<taus.length; t++) {
       int i = 0;
       for (int k=0; k<numberGrids; k++) {             // FIND CORRECT FIELDS
TO USE FOR THE HEADER
         if (gridsOut[k].verticalAxesUnitName.equals(htCoord) &&
             gridsOut[k].forecastEventTime == taus[t] &&
             gridsOut[k].data.length > 1) {            // Change to correct
value when known
```

```
            i = k;
          }
        }
        // Determine the VT of the grid
        Pattern pat = Pattern.compile("[. ]");
        String strs[] = pat.split(gridsOut[i].forecastRunTimeString);
        int cTau = gridsOut[i].forecastEventTime;
        float[] latlon = gridsOut[i].getBoundingBox();
        // CREATE THE VALID TIME OF THE GRID
        Calendar validTime = Calendar.getInstance();
        validTime.clear();
        validTime.set((int)Integer.parseInt(strs[0]),   // Set the Year
                      (int)Integer.parseInt(strs[1])-1, // Set the Month
                      (int)Integer.parseInt(strs[2]),   // Set the Date
                      (int)Integer.parseInt(strs[3]),   // Set the Hours
                      0, 0);                 // Set the Minutes and Seconds
        validTime.add(Calendar.HOUR_OF_DAY, cTau);
        int correctMonth = validTime.get(Calendar.MONTH)+1;
        String temp = ""+validTime.get(Calendar.YEAR);
        String tempYear = temp.substring(2);
        double[] zLevel = gridsOut[i].zAxisValueArray;            // Get the
zLevel information
        // OUTPUT THE HEADER INFORMATION TO THE FILE
        pw.println("FFFFFFFF");                                   // LINE #1
        pw.println("NRLCOAMPS");                                  // LINE #2
        writeRSpace(""+validTime.get(Calendar.DAY_OF_MONTH));     // LINE #3
        writeRSpace(""+correctMonth);
        writeRSpace(""+tempYear);
        writeRSpace(""+validTime.get(Calendar.HOUR_OF_DAY));
        writeRSpace("0");
```

```
writeRSpace("0");
pw.println();
writeRSpace(""+validTime.get(Calendar.DAY_OF_MONTH));      // LINE #4
writeRSpace(""+correctMonth);
writeRSpace(""+tempYear);
writeRSpace(""+validTime.get(Calendar.HOUR_OF_DAY));
writeRSpace("0");
writeRSpace("0");
pw.println();
writeRSpace(""+gridsOut[i].data[0][0].length);             // LINE #5
writeRSpace(""+gridsOut[i].data[0].length);
int zDataLen = gridsOut[i].data.length;
if (gridsOut[i].data.length > 20) {
  writeRSpace("20");
  zDataLen = 20;
} else {
  writeRSpace(""+gridsOut[i].data.length);
}
writeRSpace("0");
writeRSpace(""+strAttributeCodes3.length);
writeRSpace(""+strAttributeCodes2.length);
pw.println();
writeRSpace("0");                                          // LINE #6
writeRSpace("0");
writeRSpace("0");
writeRSpace("0");
writeRSpace("0");
writeRSpace("0");
pw.println();
writeRSpace("0");                                          // LINE #7
```

```
writeRSpace("0");
writeRSpace("0");
pw.println();
int tempv = 0;
for (int j=0; j<zDataLen; j++) {                              // LINE #8
  writeRSpace(""+form.format(zLevel[j]));
  tempv++;
  if (tempv == 6) {
    pw.println();
    tempv = 0;
  }
}
writeRSpace(""+form.format(gridsOut[i].xAxisSpacing));
writeRSpace(""+form.format(gridsOut[i].yAxisSpacing));
writeRSpace("-999999.0000");
writeRSpace("-999999.0000");
pw.println();
writeRSpace(""+form.format(latlon[1]));
writeRSpace(""+form.format(latlon[0]));
writeRSpace("0.0000");
writeRSpace("0.0000");
writeRSpace("0.0000");
writeRSpace("0.0000");
pw.println();
writeRSpace("0.0000");
pw.println();
Pattern patt = Pattern.compile("[:]");
StringBuffer param3D = new StringBuffer("");
StringBuffer units3D = new StringBuffer("");
for (int j=0; j<strAttributeCodes3.length; j++) {
```

```
          String tempSearch =
""+(String)convertTable.get(strAttributeCodes3[j]);
          if (tempSearch.equals("null")) {
            tempSearch = "junk:junk";
          }
          String pu[] = patt.split(tempSearch);
          param3D.append(writeLSpace(pu[0]));
          units3D.append(writeLSpace(pu[1]));
        }
        pw.println(param3D);
        pw.println(units3D);
        StringBuffer param2D = new StringBuffer("");
        StringBuffer units2D = new StringBuffer("");
        for (int j=0; j<strAttributeCodes2.length; j++) {
          String tempSearch =
""+(String)convertTable.get(strAttributeCodes2[j]);
          if (tempSearch.equals("null")) {
            tempSearch = "junk:junk";
          }
          String pu[] = patt.split(tempSearch);
          param2D.append(writeLSpace(pu[0]));
          param2D.append(writeLSpace(pu[1]));
        }
        pw.println(param2D);
        // FIND PRESSURE GRID
        for (int k=0; k<numberGrids; k++) {
          if (gridsOut[k].parameterName.equals("TOTAL_PRESSURE")) {
             presData = gridsOut[k].data;
          }
        }
```

```
        // WRITE OUT 3D GRIDS


        for (int k=0; k<numberGrids; k++) {
          if (gridsOut[k].forecastEventTime == taus[t]) {
            tempv = 0;
            float[][][] gridData = gridsOut[k].data;
            int zT=gridsOut[k].data.length;
            if (gridsOut[k].data.length > 20) {zT=20;}
            for (int zt = 0; zt<zT; zt++) {
              for (int yt = 0; yt<gridsOut[k].data[0].length; yt++) {
                for (int xt = 0; xt<gridsOut[k].data[0][0].length; xt++) {
                  if (gridsOut[k].parameterName.equals("TOTAL_PRESSURE")) {
                    Double sigma = new Double(zLevel[zt]);
                    float mHeight = (((float)34800.0 -
gridsOut[k+1].data[0][yt][xt]) * sigma.floatValue()/(float)34800.0);
                    gridData[zt][yt][xt] = mHeight*(float)9.806/(float)9.8;
                  }
                  writeRSpace(""+form.format(gridData[zt][yt][xt]));
                  tempv++;
                  if (tempv == 6) {
                    tempv=0;
                    pw.println();
                  }
                }
              }
            }
          if (tempv != 0) {
            pw.println();
          }
          }
```

```
     }
   }
   pw.close();
   fos.close();
 } catch (Exception e) {
   System.out.println("The error in writing the file was:"+e);
 }
}

/**
 * Write output data in Right Justified 12 space format required.
 * Writes out the information about the grids returned from TEDServices
 * @param inString Input String to be formatted
 * @return outString The formatted String
 */
public void writeRSpace(String inString) {
  StringBuffer outString = new StringBuffer("           !");
  outString.replace(12-inString.length(),12,inString);
  outString.append(" ");
  pw.print(outString);
}

/**
 * Write output data in Left Justified 9 space format required.
 * Writes out the information about the grids returned from TEDServices
 * @param inString Input String to be formatted
 * @return outString The formatted String
 */
public StringBuffer writeLSpace(String inString) {
  StringBuffer outString = new StringBuffer("         ");
```

```
  outString.replace(0,inString.length(),inString);
  return outString;
}

/**
 * Collect a web page for parsing.
 * Collects the TEDServices Thin Client information for data checking
 * @param whatPage URL to be collected
 * @return tPageContent The StringBuffer containing the entire page
 */
public StringBuffer getPage(String whatPage) {
  StringBuffer tPageContent = new StringBuffer("");
  try {
    int c;
    URL TEDSurl = new URL(whatPage);
    URLConnection checking = TEDSurl.openConnection();
    InputStream input  = checking.getInputStream();
    int i = checking.getContentLength();
    while (((c=input.read()) != -1)) { // && (--i > 0)) {
      tPageContent.append((char) c);
    }
    input.close();
  } catch (Exception e) {
    System.out.println("ERROR"+e);
  }
  return tPageContent;
}

/**
 * Find out Which Taus need to be retrieved.
```

```
   * Checks TEDServices and decides which data should be pulled based upon
   * the inString and the available data
   * @param inString The date formatted as 2003.09.19.00.00
   * @return tauList An integer array of the taus to be collected from
TEDServices
   */
  public int[] tauCalc(String inString) {
    String cDTG = "";
    int[] DTGTList = new int[]{0};
    try {
      //  Get most recent run time
      String runTime =
"http://"+database+"/servlet/TEDSThinClientServlet?storeSource=VNE&classificati
on=ATMOSPHERIC_FORECAST";
      StringBuffer pageContent = getPage(runTime);
      int lastOption = pageContent.lastIndexOf("<option value=");
      cDTG = pageContent.substring(lastOption+15,lastOption+38);
      String lastDTG =
pageContent.substring(lastOption+15,lastOption+38).replace(' ','+');
      //  Get taus
      TreeSet tauSet = new TreeSet();
      String tauTime =
"http://"+database+"/servlet/TEDSThinClientServlet?storeSource=VNE&classificati
on=ATMOSPHERIC_FORECAST&forecastRunTime="+lastDTG;
      pageContent = new StringBuffer("");
      pageContent = getPage(tauTime);
      int option = 1, optionEnd, lastFind = 1;
      while (option > 0)  {
        option = pageContent.indexOf("<option value=", lastFind);
        optionEnd = pageContent.indexOf("\"", option+15);
        lastFind = optionEnd;
        if (option > 0 ) {
```

```
tauSet.add(Integer.valueOf(pageContent.substring(option+15,optionEnd)));
        }
      }
      DTGTList = new int[tauSet.size()];
      Iterator itr = tauSet.iterator();
      int dumi = 0;
      while(itr.hasNext()) {
        DTGTList[dumi] = ((Integer)itr.next()).intValue();
        dumi++;
      }
    } catch (Exception e) {
      System.out.println("ERROR"+e);
    }
    // calc the tau to start counting with
    Pattern pat = Pattern.compile("[. ]");
    // CREATE THE VALID TIME OF THE INCIDENT
    String strs[] = pat.split(inString);
    Calendar eventTime = Calendar.getInstance();
    eventTime.clear();
    eventTime.set((int)Integer.parseInt(strs[0]), // Set the Year
                  (int)Integer.parseInt(strs[1]), // Set the Month
                  (int)Integer.parseInt(strs[2]), // Set the Date
                  (int)Integer.parseInt(strs[3]), // Set the Hours
                  0);                             // Set the Minutes
    // CREATE THE VALID TIMES FOR EACH TAU AND CHECK IF GREATER THAN INCIDENT
TIME
    Stack st = new Stack();
    String strs2[] = pat.split(cDTG);
    Calendar tauTime = Calendar.getInstance();
```

```
  for (int i=DTGTList.length-1; i>-1; i--) {
    tauTime.clear();
    tauTime.set((int)Integer.parseInt(strs2[0]), // Set the Year
                (int)Integer.parseInt(strs2[1]), // Set the Month
                (int)Integer.parseInt(strs2[2]), // Set the Date
                (int)Integer.parseInt(strs2[3]), // Set the Hours
                0);                              // Set the Minutes
    tauTime.add(Calendar.HOUR, DTGTList[i]);
    if (tauTime.after(eventTime)) {
      st.push(new Integer(DTGTList[i]));
    }
  }

  // Convert the list back to an integer array for use in the Grid3D call
  int ll = st.size();
  int[] tauList = new int[ll];
  for (int i=0; i<ll; i++) {
    Integer a = (Integer)st.pop();
    tauList[i] = a.intValue();
  }
  return tauList;
}



public static void main(String[] args) {

  if (args.length == 5){
```

```
      float[] inBox = new
float[]{Float.parseFloat(args[0]),Float.parseFloat(args[1]),Float.parseFloat(ar
gs[2]),Float.parseFloat(args[3])};
      new JEMWeather(inBox, args[4]);
    } else {
      System.out.println("Usage:   JEMWeather <North> <South> <East> <West>
<YYYY.MM.DD.HH.MM>");
      System.out.println("Example: JEMWeather 33.0 32.5 -117.0 -117.5
2003.09.09.14.00");
      System.exit(1);
    }
  }
}
```

LIST OF REFERENCES

Defense Threat Reduction Agency. *Hazard Prediction and Assessment Capability User Guide.* February 2003.

Integrated Chemical and Biological Defense Research, Development and Acquisition Plan. *Chemical & Biological Point Detection Decontamination Information Systems*. April 2003.

Joint Effects Model Integrated Product Team. [http://www.jointeffectsmodel.org/JEM/Portal/Virtual%20Prototype/]. September 2003.

Naval Research Laboratory – Stennis Space Center. *TEDServices 2.2 Application Programming Interface Documentation*. September 2003.

Oceanographer of the Navy (N096). (2002, March). *Naval Oceanography Program Operational Concept*. Washington, DC: U.S. Government Printing Office.

Office of the Federal Coordinator for Meteorological Services and Supporting Research (OFCM). (2002, August). *Atmospheric Modeling of Releases from Weapons of Mass Destruction: Response by Federal Agencies in Support of Homeland Security*. Washington, DC: U.S. Government Printing Office.

Ozone Database Project. “What is it?.” [http://www.ozone-db.org/frames/home/what.html]. September 2003.

Schildt, Herbert. *Java 2: The Complete Reference, Fifth Edition*. McGraw-Hill, 2002.

Statement by Dr. Anna Johnson-Winegar. Deputy Assistant to the Secretary of Defense for Chemical and Biological Defense. to the House Government Reform Committee, Subcommittee on National Security, Emerging Threats, and International Relations. 2 June 2003.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Prof. Neil Rowe
   Naval Postgraduate School
   Monterey, CA

4. Prof. Chuck Wash
   Naval Postgraduate School
   Monterey, CA

5. Captain Joe Atangan, USN (N5)
   Commander Naval Meteorology and Oceanography Command
   Stennis Space Center, MS

6. John Shea
   Space and Warfare Command, PMW 150
   San Diego, CA

7. John Raby
   U.S. Army Research Laboratory
   White Sands Missile Range, NM

8. Commander Bruce Morris, USN
   Naval Pacific Meteorology and Oceanography Center – San Diego
   San Diego, CA